

Learning User and Item Representations
from Heterogeneous Feedback
for Accurate and Explainable Recommendation

Lei Liu

A thesis submitted for the degree of

Doctor of Philosophy

The Australian National University



Australian
National
University

September 28, 2026

© Copyright by Lei Liu, 2026

All Rights Reserved

Declaration of Authorship

I hereby declare that the work undertaken in this thesis has been conducted by me alone, except where indicated in the text. I conducted this work between November 2020 and April 2025, during which period I was a PhD student at the Australian National University. This thesis, in whole or any part of it, has not been submitted to this or any other university for a degree.

A handwritten signature in black ink, reading "Lei Liu". The script is cursive and fluid, with the first name "Lei" and last name "Liu" clearly distinguishable.

Lei Liu
September 28, 2026

Acknowledgements

Pursuing a PhD has been a remarkable journey, filled with curiosity, learning, self-discovery, and its fair share of challenges. Along the way, I have been incredibly fortunate to receive support from so many people, without whom I could not have reached this point.

First and foremost, I am deeply grateful to my primary supervisor, Professor Penny Kyburz. Her steady support, guidance, and encouragement have been essential throughout my PhD. Penny gave me generous access to research resources, including substantial GPU resources, and helped me gain access to the School's computing clusters when more computational power was needed. She also encouraged us to take part in a wide range of academic activities, such as showcases, research retreats, and seminars, which greatly broadened my academic experience.

Her supervision has been both rigorous and supportive. She respected my research ideas while giving me the guidance I needed whenever I went off track. Her detailed feedback on my writing, down to individual word choices and sentence structure, greatly improved the way I communicate my research. Penny's efficiency and precision have left a lasting impression on me, and I know I still have a long way to go before I can follow her example. More than once, she comforted me when experiments failed, was patient with my delays and mistakes, and helped me find my way forward when I felt stuck. I often wonder whether, if I were in her position, I could show a student the same level of care and patience.

I am also sincerely grateful to my associate supervisors, Professor Lexing Xie, Dr Jooyoung Lee, and Dr Priscilla Kan John, for their valuable insights and suggestions, especially during my thesis proposal review. I am especially thankful to Dr Xinyu Hu, whose thoughtful advice helped shape the direction of my project and who also shared her own reflections on the PhD journey. I am equally grateful to my master's supervisor, Professor Chen Yang, for mentoring me during my early years in research. His guidance sparked my interest in research and gave me the foundation that eventually led me to pursue a PhD.

I have also had the privilege of being part of a research group that was not only intellectually stimulating but also warm and fun. Every colleague inspired me with their dedication, intelligence, and kindness.

To Matthew Aitchison, thank you for the many hours you spent helping me troubleshoot problems during my first research project. Our conversations, whether about research or the philosophy of science, often gave me new perspectives and ideas.

To Runze Tang, our many discussions about project details and the time we spent talking through failed experiments were a real source of inspiration. Sharing the excitement of breakthroughs with you made the journey all the more rewarding.

To Mahsum Daiiani, Michelle Adiwangsa, Yufei Cao, and Karla Kelly, your projects have been fascinating and meaningful. Taking part in your carefully designed experiments broadened my understanding of creativity and rigour in scientific research. I am especially thankful to Mahsum for organising memorable group picnics, to Michelle for going to the cinema with me, to Yufei for getting me into fitness, and to Karla for the endless supply of puzzles in the office. On anxious evenings, putting those pieces together brought me a comforting sense of order.

To Wanqi Zhao, thank you for being by my side through both the happy and difficult moments. I will always remember our trip to Sydney to see Jay Chou's concert, the dazzling Drone Sky Show in Canberra, and all our conversations in between. Your calm and analytical way of thinking helped me through many difficult decisions. Watching your progress from a complete beginner to a competitive tennis player taught me a great deal about the power of planning and perseverance.

Canberra has been a quiet and beautiful city to live in. Coming from a place with scorching summers and damp winters, I have grown to love its dry winters and mild summers. Here, I found a group of friends with whom I felt completely at ease, and they filled my three years in Canberra with joy and connection.

To Fangqin Chen, Shihao Jiang, Jinyin Hu, and Morris Chang, every plan we made together gave me something to look forward to. From Floriade and Christmas in July to the Royal Canberra Show, we made memories all over the city. We went hiking, picked cherries and chestnuts, and travelled together to explore Australia's landscapes, skies, and seas. We celebrated festivals, experimented with new recipes, and worked our way through every restaurant on our list. I will always cherish these memories and the kindness and support you have given me.

To my dear friends in China, Yanmin Jiao and Yanan Wu, thank you for always being there despite the distance and time difference. Your support has meant more to me than I can say.

This PhD journey has not been without its difficulties. I spent more than a year at home during the pandemic and completed the final stage of my thesis remotely.

Through it all, my family's love and support never wavered.

To my mother, Zhengrong Lei, and my father, Chuanbing Liu, thank you for always listening, trusting my decisions, and being there whenever I needed you.

To my grandmothers, Liangfeng Chen, whom I miss dearly, and Zeju Wu, your words gave me the courage to pursue the life I wanted.

To Liuliu, the stray kitten adopted by my mum, you may be aloof and mischievous, and yes, you did chew through the corner of my brand-new monitor, but your quiet company brought me comfort through many long days. Watching you sleep belly-up and completely carefree often eased my stress in ways I never expected.

Lastly, I want to thank myself for not giving up. Just outside our office stands a tall tree. It remains steady through sunshine, wind, and rain. On many quiet nights, I stood by the window looking at it and wondering what the future would bring. The PhD journey was not always bright, and there are still things I wish I had done differently. I have come to realise that my desire to do everything perfectly often came from a belief that I had to be perfect to be accepted, respected, or loved. I also used to think that asking for help was a sign of weakness rather than a way of connecting with others. These thoughts held me back more than I realised. Yet I am grateful that this PhD journey has helped me begin to understand myself more deeply.

I hope the road ahead will lead me to become not only a better version of myself, but also the person I have always wanted to be.

Abstract

Representation learning is central to recommender systems because the information encoded in user and item representations directly affects how effectively user preferences and item characteristics can be modelled. Recommender systems increasingly learn from heterogeneous feedback, including behavioural interactions, numerical ratings, and textual reviews. These feedback sources provide complementary information, but they also raise an important representation learning problem: the same user or item may need to be represented differently depending on the type of feedback being modelled or the particular interaction for which a prediction is made. This thesis investigated this problem from the perspective of accurate and aspect-level explainable recommendation.

The thesis examined two related research questions. The first concerned how implicit and explicit feedback could be integrated while preserving their different meanings and allowing information to be exchanged between their respective user and item representations. The second concerned how user and item representations could depend on their current interaction counterpart in aspect-level recommendation.

To investigate these questions, the thesis adopted a polymorphic representation learning perspective, under which the same entity can have different representations across feedback types or interaction counterparts. For the first research question, we developed Interdependent Embedding Learning (IEL), a GCN-based framework that maintained separate user and item representations for implicit and explicit feedback. IEL used intra-graph and selected inter-graph message passing to allow these feedback-specific representations to exchange collaborative information and was evaluated for both item ranking and rating prediction. For the second research question, we developed Dynamic Representation Learning (DRL), which combined general collaborative representations learned from ratings with aspect-sentiment information extracted from textual reviews for the current interaction. DRL constructed an item-dependent user representation and a user-dependent item representation for rating and positive-aspect prediction. A prompt-based large language model (LLM) procedure was used to transform reviews into structured aspect-sentiment information for this study.

Offline experiments on public Amazon product-review datasets provided empirical support for both approaches. IEL improved recommendation performance relative to the compared unified recommendation baselines, with the strongest and most consistent improvements observed for item ranking. DRL achieved lower rating-prediction

MSE than the compared baselines across all three evaluated datasets, with an average reduction of 5.1% relative to the strongest compared baseline. It also improved Hit@1 for positive-aspect prediction by an average of 4.8%. The latter evaluation measured the model’s ability to predict aspects subsequently appearing in held-out reviews and therefore served as an operational measure of aspect-level explanation.

The main contribution of this thesis is a unified representation-learning perspective that connects feedback-dependent and interaction-dependent representations. The IEL and DRL studies demonstrate how heterogeneous feedback can be preserved and integrated without requiring a single invariant representation of each user or item. In addition, the thesis constructed three derived Amazon review datasets containing LLM-extracted aspect-sentiment information for aspect-level recommendation research. Overall, the findings suggest that allowing user and item representations to vary according to the feedback type and the interaction being modelled provides a useful direction for learning from heterogeneous feedback in accurate and explainable recommender systems.

Contents

Declaration of Authorship	i
Acknowledgements	ii
Abstract	v
1 Introduction	1
1.1 Implicit and Explicit Feedback	3
1.2 Recommendation Methods	5
1.2.1 Content-Based Methods	6
1.2.2 Collaborative Filtering (CF)	7
1.2.3 Matrix Factorisation (MF)	8
1.2.4 Neural CF	10
1.2.5 Graph Neural Network (GNN)-Based Methods	11
1.2.6 Review-Based Methods	12
Topic-Level Models	13
Review-Level Models	14
Aspect-Level Models	16
1.3 Research Problem and Challenges	19
1.4 Research Questions	21
1.5 Research Methodology	22
1.6 Research Contributions	23
1.7 Thesis Outline	24
2 Literature Review	26
2.1 Multiple Representation Learning	28
2.1.1 Multiple Feedback	28
2.1.2 Single Feedback	32
2.2 Target-Aware and Interaction-Dependent Representation Learning	35

2.3	Aspect-Level Recommendation Models	40
2.4	Explanation Methods and Forms	45
2.5	Evaluation of Explanations	47
2.5.1	Online Evaluations	47
2.5.2	User Studies	48
2.5.3	Offline Formula-Based Methods	48
2.5.4	Case Studies	51
3	IEL: Interdependent Embedding Learning	53
3.1	Related Work	57
3.1.1	Rating Prediction	57
3.1.2	Ranking Recommendation	57
3.1.3	Unified Tasks	59
3.1.4	Summary and Research Gap	60
3.2	Preliminaries and Model Motivation	61
3.2.1	Matrix Factorisation	62
3.2.2	Bayesian Personalised Ranking	62
3.2.3	Graph Convolutional Networks	63
	Neighbour Aggregation.	63
	Node Update.	64
3.2.4	Model Architecture Motivation	64
3.3	Interdependent Embedding Learning	67
3.3.1	Intra-Feedback GCN	68
3.3.2	Inter-Feedback GCN	69
3.3.3	Layer Combination and Model Prediction	71
	Layer Combination	71
	Model Prediction	72
3.3.4	Model Optimisation	72
3.3.5	Design Decisions and Other Architectural Choices	73
	Using Attention Mechanism Instead of Linear Message Passing .	73
	Adopting Non-Linear Prediction for Implicit Scores	73
	Node Learning versus Edge Learning	74
3.4	Experiments	74
3.4.1	Experimental Setup	74
3.4.2	Item Ranking Performance	78

3.4.3	Rating Prediction Performance	79
3.4.4	Impact of Number of Layers	80
	Item Ranking	81
	Rating Prediction	81
3.5	Discussion	82
3.5.1	Limitations and Future Work	85
3.6	Conclusions	87
4	DRL: Dynamic Representation Learning	89
4.1	Related Work	93
4.1.1	Aspect-Level Models	93
4.1.2	Target-Aware and Interaction-Dependent Representations	95
4.1.3	Explainable Recommendation Models	96
4.2	Preliminaries	97
4.2.1	User-Aspect Attention Matrix	98
4.2.2	Item-Aspect Quality Matrix	98
4.3	Proposed Model	98
4.3.1	Rating-Based GNN Layer	100
	User-Side GNN	101
	Item-Side GNN	101
	Layer Combination	101
4.3.2	Dynamic Representation Layer	102
	Edge Encoder	102
	Dynamic User Embedding	104
	Dynamic Item Embedding	104
4.3.3	Prediction Layer	105
	Rating Prediction	105
	Aspect-Level Review Prediction	105
4.3.4	Model Training	106
4.4	Explanation Format and Evaluation	107
4.4.1	Explanation Format	107
4.4.2	Explanation Evaluation	108
4.5	Experiments	109
4.5.1	Extraction of Aspects and Corresponding Sentiment Polarity	111
4.5.2	Aspect Filtering	114

4.5.3	Aspect Embedding Generation	114
4.5.4	Baselines and Parameter Settings	115
4.5.5	Rating Prediction Evaluation	117
4.5.6	Explainability Evaluation	118
	Offline Formula-Based Method	120
	Case Study	123
4.5.7	Effect of Varying GNN Layer Depths	128
4.5.8	Ablation Study	129
	Remove GNN Layer	129
	Retain Target Edge in Dynamic Representation Learning	129
	Disable Semantic Aspect Embeddings	129
	Remove Dummy Aspect	130
4.6	Discussion	131
4.6.1	Limitations and Future Work	135
4.7	Conclusions	138
5	Discussion	139
5.1	Key Findings of IEL	140
5.2	Key Findings of DRL	140
5.3	Addressing the Research Questions	141
5.4	Thesis Significance and Broader Impact	142
5.5	Limitations and Future Work	143
6	Conclusions	146
	Bibliography	148

List of Figures

1.1	Two 4-star reviews on the same product reveal distinct user concerns. <i>Source: Amazon.com</i>	4
3.1	An illustration of three possible sources of information for message aggregation in a dual-graph setting. The implicit and explicit representations of user i and item j are denoted by implicit user u_i , implicit item z_j , explicit user q_i , and explicit item v_j , respectively. Taking the implicit item z_j as an example, its candidate information sources include (1) neighbouring implicit users on the same graph, (2) its explicit counterpart v_j , and (3) neighbouring explicit users on the other graph. The figure illustrates the architectural choices that arise when learning feedback-specific representations across the implicit and explicit graphs; not all candidate pathways are necessarily used in the final IEL architecture.	65
3.2	The overall framework of IEL, a GCN-based recommendation model designed to integrate implicit and explicit feedback. Each user and item is represented by two distinct embeddings for the two feedback types: u_i and q_i for user i , and z_j and v_j for item j . Implicit and explicit feedback are modelled via message passing within and across graphs. Blue arrows denote the direction of message aggregation. Specifically, z_j integrates information from explicit user embeddings, while q_i aggregates messages from implicit item embeddings, enabling inter-graph information transfer.	67
3.3	Ranking performance of IEL with varying GCN layers on the Amazon-Office dataset.	81
3.4	Ranking performance of IEL with varying GCN layers on the Amazon-Video Games dataset.	82

3.5	Rating prediction performance of IEL with varying GCN layers across different datasets.	82
4.1	An example review from Amazon.com, along with two extracted triples (aspect, opinion phrase, sentiment polarity). Here, sentiment polarity is assigned a value of +1 for positive sentiment and -1 for negative sentiment.	90
4.2	The architecture of DRL. (i) A rating-based GNN layer that generates general embeddings for users and items. (ii) A dynamic representation layer that utilises an edge encoder to transform aspect information into edge embeddings and an attention mechanism to generate interaction-dependent embeddings. (iii) A prediction layer that integrates the user and item interaction-dependent embeddings to predict the rating and identify the positive aspects.	99
4.3	Python code for generating prompts to extract or infer structured (aspect, sentiment score) pairs from textual reviews. The function takes the current dataset, examples, and the target review as inputs, and encodes detailed task instructions, including four explicit rules for the model to follow.	111
4.4	Explainability performance on the Office dataset with varying numbers of predicted aspects. Subplots (A), (B), and (C) report Hit, Recall, and Precision respectively. For DRL, exact values at $N = 1$ and $N = 2$ are annotated on the curves.	119
4.5	Explainability performance on the Video Games dataset with varying numbers of predicted aspects. Subplots (A), (B), and (C) report Hit, Recall, and Precision respectively. For DRL, exact values at $N = 1$ and $N = 2$ are annotated on the curves.	119
4.6	Explainability performance on the CDs & Vinyl dataset with varying numbers of predicted aspects. Subplots (A), (B), and (C) report Hit, Recall, and Precision respectively. For DRL, exact values at $N = 1$ and $N = 2$ are annotated on the curves.	120
4.7	Rating prediction performance of DRL at different GNN layers across different datasets.	128

List of Tables

2.1	Aims of explanations in recommender systems (adapted from Tintarev (2007)).	27
3.1	Models utilising implicit and explicit feedback.	58
3.2	Summary of notations.	68
3.3	Statistics of the datasets.	74
3.4	Overall comparison of ranking performance. The best results are highlighted in bold, and the second-best results are underlined.	79
3.5	Overall comparison of rating prediction performance. The best results are highlighted in bold, and the second-best results are underlined. . . .	80
4.1	Representative models utilising reviews.	94
4.2	Top 10 of filtered aspects in each dataset.	103
4.3	Statistics of the datasets.	113
4.4	Statistics of original reviews and aspects in each dataset.	113
4.5	Statistics of filtered aspects in each dataset.	114
4.6	Challenging aspects and their adjustments.	115
4.7	Overall comparison of rating prediction performance. The best results are highlighted in bold, and the second-best results are underlined. . . .	118
4.8	Overall comparison of explanation performance. In this evaluation, the ground truth consists of reviews associated with actual ratings greater than 3. The best results are highlighted in bold, and the second-best results are underlined.	121
4.9	Overall comparison of explanation performance. In this evaluation, the ground truth consists of reviews associated with predicted ratings greater than 3. The best results are highlighted in bold, and the second-best results are underlined.	123
4.10	Predicted rating and top-5 positive aspects versus actual rating and user review (Office Dataset).	124

4.11 Predicted rating and top-5 positive aspects versus actual rating and user review (Video Games Dataset).	125
4.12 Predicted rating and top-5 positive aspects versus actual rating and user review (CDs & Vinyl Dataset).	127
4.13 Explainability performance of DRL at different GNN layers across different datasets.	129
4.14 Rating prediction performance of DRL and its four variants. The best results are highlighted in bold, and the second-best results are underlined.	131
4.15 Explainability performance of DRL and its four variants. The best results are highlighted in bold, and the second-best results are underlined.	131

Chapter 1

Introduction

Recommender systems have become widely used tools for information filtering across online retail, streaming services, and social media platforms (Wu et al., 2022; De Campos et al., 2024). Personalised recommender systems learn from users’ historical feedback, including implicit interactions, explicit ratings, and textual reviews, to model users’ preferences and item characteristics. A central technical problem is therefore how to construct informative representations of users and items from the available feedback. Predictive accuracy has historically been a major objective of recommender systems, but other objectives, including explainability, have also received increasing attention. Explanations can help users understand the information associated with a recommendation, although their perceived usefulness, persuasiveness, and effects on trust depend on how explanations are designed and evaluated (Wardatzky et al., 2025).

Explainability broadly concerns the question of *why* an item is recommended, in contrast to *how* the underlying algorithm works, which falls under interpretability (Pan et al., 2020). It is worth noting that these two terms have been used interchangeably in some prior work, such as Liu et al. (2021) and Zhang et al. (2021). Some research (Tintarev and Masthoff, 2012) has argued that user-centred explanations need not expose all internal mechanisms of the model. Instead, they should fulfil specific explanatory purposes, such as supporting efficient decision-making or enhancing persuasive effectiveness (Tintarev, 2007), which in turn can encourage users to engage with the system more consistently (Hulstijn et al., 2023). Accordingly, this thesis uses explainability in the former sense and focuses on user-facing reasons for why an item is recommended, rather than on interpreting the internal operation of the recommendation algorithm.

A fundamental objective of recommender systems is to infer user-item relationships from observed interactions. This approach emerged in the mid-1990s, when researchers began addressing recommendation problems primarily based on **explicit**

feedback, particularly in the form of numerical ratings (Adomavicius and Tuzhilin, 2005). Early research focused on **rating prediction**, aiming to estimate a user's rating (e.g., from 1 to 5 stars) for a specific item with high accuracy (Su and Khoshgoftaar, 2009). Classic **Collaborative Filtering (CF)** methods performed well in this context by comparing users' explicit ratings to identify either similar users or similar items. As the field of recommendation evolved, however, both researchers and practitioners recognised that **item ranking**, which prioritises the most relevant items, was often more suitable for real-world applications. The growing prevalence of **implicit feedback**, such as clicks, add-to-cart actions, and purchases, made this shift especially relevant. In ranking-oriented tasks (also known as top- N recommendation), the goal is to place a user's preferred items near the top of the recommended list.

Subsequently, various methods such as content-based recommendation, matrix factorisation, and neural network-based approaches emerged to accommodate different types of feedback and recommendation tasks. **Content-based recommendation** relies on item features (e.g., text descriptions and tags) and can be adapted for both explicit and implicit feedback (Pazzani and Billsus, 2007), although early implementations primarily utilised explicit signals to model user-item relevance. **Matrix Factorisation (MF)** gained prominence during the Netflix Prize competition¹ (Wu et al., 2023). It initially focused on rating prediction using explicit feedback and was later extended to handle implicit feedback, typically through a pairwise ranking loss (Rendle et al., 2009). As recommendation models became increasingly sophisticated, **Neural Network-based approaches** were introduced. These methods employ architectures such as Autoencoders (AE) and Convolutional Neural Networks (CNN), enabling flexible modelling of both explicit and implicit feedback (Zhang et al., 2019b). Given that user-item interaction in recommender systems can naturally be represented as a bipartite graph, where users and items serve as two distinct types of nodes and observed interactions form edges between them, **Graph Neural Network (GNN)-based methods** have received increasing attention (Wu et al., 2022). This technique also supports the modelling of both explicit and implicit feedback across a range of recommendation tasks.

Across these recommendation paradigms, **representation learning** concerns how information about users and items is encoded into forms that can be used for preference prediction. Traditional approaches often relied on explicit user/item profiles or manually constructed features, whereas MF and modern neural methods learn latent

¹https://en.wikipedia.org/wiki/Netflix_Prize

vector representations from user-item interaction data (Li and Zhao, 2021). More recent recommender systems increasingly enrich these representations with additional information, such as textual reviews, knowledge graphs, and multi-modal item content (Hasan et al., 2025; Wu et al., 2022). The quality of these representations directly affects what information is available to the matching or prediction function. This thesis focuses primarily on this representation-learning stage: rather than designing increasingly complex matching functions, it investigates how user and item representations can preserve and integrate information from heterogeneous feedback and how these representations can vary according to the current interaction context.

In the following sections, we first distinguish implicit and explicit feedback and explain their relationships with different recommendation objectives (Section 1.1). We then review major recommendation approaches with particular attention to how they construct user and item representations (Section 1.2). In general, these approaches differ in their ability to provide explanations: methods based on directly identifiable content features or neighbourhood relationships can often expose information that contributed to a recommendation, whereas latent representation models such as MF, neural CF, and many GNN-based approaches generally require additional information or mechanisms to produce user-facing explanations.

1.1 Implicit and Explicit Feedback

Implicit feedback records users' behavioural interactions with items and can often be collected automatically at scale (Chen et al., 2017). In common binary implicit-feedback formulations, observed interactions are encoded as 1 and unobserved interactions as 0. An observed interaction indicates engagement with an item but does not directly reveal the strength or direction of the user's evaluation, while an unobserved interaction does not necessarily indicate dislike (Hu et al., 2008). Implicit feedback is therefore commonly modelled as interaction occurrence and used for ranking-oriented recommendation, where the objective is to place relevant items near the top of a recommendation list. Offline evaluation commonly employs information retrieval metrics such as Precision@ N , Recall@ N , and NDCG@ N (Su and Khoshgoftaar, 2009). Another common task based on implicit feedback is click-through rate (CTR) prediction, which estimates the probability that a user will click a candidate item or advertisement. CTR prediction is an important large-scale learning problem in industrial recommender and advertising systems (McMahan et al., 2013), and models for this task are commonly

Cordless Wet Dry Vacuum Cleaner, One-Step Cleaning for Hard Floors, Auto Self-Cleaning, Edge Clean, LED Display, Long Runtime

	(1)	 Reviewed on 26 November 2024 I give it 4 stars just because it is a bit heavy to use after a while but, it's a great vacuum mop.
	(2)	 Reviewed on 5 October 2024 Really easy to set up and use, the floors feel nice and clean after. But must clean and empty out right after use as it will smell if not.

FIGURE 1.1: Two 4-star reviews on the same product reveal distinct user concerns. *Source: Amazon.com*

evaluated using metrics such as AUC and log loss (Guo et al., 2017). The present thesis, however, focuses on ranking-oriented recommendation when modelling implicit feedback, and the implicit-feedback view investigated in Chapter 3 is therefore evaluated through top- N ranking rather than CTR prediction.

In contrast, explicit feedback provides more direct evaluative information about observed user-item relationships. Numerical ratings, for example, indicate not only that an interaction occurred but also the degree of preference expressed by the user. Such feedback usually requires additional user effort and can therefore be sparser than automatically recorded behavioural interactions (Balabanović and Shoham, 1997). Rating prediction models estimate unknown rating values and are commonly evaluated using error-based metrics such as RMSE and MAE (Steck, 2013).

Implicit and explicit feedback thus provide complementary information about user-item relationships rather than simply differing in feedback quality. A concrete example is an e-commerce platform in which views, clicks, add-to-cart actions, and purchases provide behavioural signals, while star ratings and likes/dislikes provide explicit evaluative feedback from a subset of users. Such heterogeneous interaction histories have long been present in commercial recommendation settings; for example, Amazon's recommendation system was described as exploiting information including viewed, purchased, and rated items (Linden et al., 2003).

A common limitation of both ratings and behavioural signals is that they only reflect users' overall attitudes towards items, without revealing the underlying reasons behind their choices (Guan et al., 2019). In contrast, **textual reviews** provide detailed insights into users' perceptions and sentiments towards specific item features (He et

al., 2015; Pan et al., 2022). Even when users assign the same rating to an item, their motivations and concerns may differ significantly (Hou et al., 2019). For instance, as illustrated in Figure 1.1, two users gave the same vacuum cleaner a 4-star rating, but for different reasons. One user praised its performance but found it somewhat heavy, while the other appreciated its ease of setup but noted the necessity for frequent cleaning. This example highlights how identical ratings can hide different user concerns and preferences. Consequently, textual reviews provide an additional source of fine-grained information for modelling user preferences and item characteristics.

Reviews often contain users' reasons for their evaluations, including opinions towards individual item characteristics, which makes them particularly useful for review-based and explainable recommendation (McAuley and Leskovec, 2013; Hasan et al., 2025). Recent review-based recommendation research continues to investigate how such textual information can be combined with interaction and rating signals to improve user/item representation learning and provide more informative recommendation outputs (Hasan et al., 2025).

1.2 Recommendation Methods

Recommendation methods differ not only in how they compute preference scores, but also in how they represent users and items. Early content-based (Section 1.2.1) and CF methods (Section 1.2.2) relied on directly observable content features or interaction histories. MF (Section 1.2.3) introduced learned latent representations, while neural (Section 1.2.4) and graph-based methods (Section 1.2.5) subsequently enabled increasingly flexible representation learning from interactions and auxiliary information. This progression improved the capacity of recommender systems to capture complex preference patterns but also made the relationship between observed feedback and learned representations less directly interpretable.

Textual reviews provide one way to enrich these representations with more fine-grained information. Binary interactions indicate whether a user engaged with an item, whereas overall ratings indicate the strength of the user's evaluation. However, both provide limited information about the specific item characteristics underlying the user's behaviour or evaluation. Reviews can reveal which aspects users discuss and how they evaluate them, providing complementary information for modelling both users and items (Sun et al., 2020). They can also provide human-readable evidence

associated with users' evaluations. Review-based recommendation methods are introduced in Section 1.2.6.

1.2.1 Content-Based Methods

Traditional content-based methods were originally designed to recommend items containing textual information, such as websites and news articles (Adomavicius and Tuzhilin, 2005). The recommendation outcome was determined solely by matching the user profile with the content of candidate items (Lops et al., 2011), where the content was typically represented using keywords (Adomavicius and Tuzhilin, 2005). Early content-based approaches, such as vector-space models, probabilistic models, and decision trees, often relied on manually engineered features (Raza et al., 2024). For example, in the vector-space model, an item is represented as a weighted vector of word features, where each dimension corresponds to a term in the vocabulary, and term weights are computed using methods such as term frequency-inverse document frequency (TF-IDF) (Salton and Buckley, 1988). A user's profile is constructed by analysing the keywords of items they have previously interacted with (Balabanović and Shoham, 1997). The similarity between the user profile and item content is then computed using heuristic measures such as cosine similarity (Turney and Pantel, 2010), and items with the highest similarity scores are recommended, ensuring alignment with the user's historical interests.

One of the key advantages of content-based methods is their transparency. The system's behaviour can be explained by explicitly listing the content features that contributed to a recommendation (Lops et al., 2011). For example, when a decision tree is used with a limited number of structured features, it is straightforward to identify the most informative features that define the decision boundaries (Pazzani and Billsus, 2007). The overlapping keywords between the user profile and the item content can also serve as an explanation for the recommendation.

Despite their simplicity and interpretability, traditional content-based methods have several inherent limitations. First, they typically rely on manually designed or heuristic features to represent users and items, which may fail to capture deeper semantic relationships. Second, heuristic similarity metrics operate on surface-level content representations and often do not reflect the true semantic relevance of items to users, potentially leading to suboptimal performance.

Modern content-based recommendation increasingly replaces manually engineered features with learned representations derived from text, images, audio, and other content modalities. Neural encoders can capture semantic information that is difficult to represent through shallow keyword vectors, thereby providing richer user and item features for recommendation (Zhang et al., 2019b). This development illustrates the broader shift towards representation learning introduced earlier in this chapter. The resulting latent representations can improve predictive modelling but are generally less directly interpretable than explicit content features. One way to retain useful semantic information while supporting explanation is to identify fine-grained, human-readable features that contributed to the learned preference representation. Review-based recommendation methods follow this general direction by combining learned representations with information extracted from users' textual feedback.

1.2.2 Collaborative Filtering (CF)

The concept of collaborative filtering is based on the idea of predicting a target user's behaviour by leveraging the collaborative behaviours of their closest "peers" (Wu et al., 2023). Typically, these systems do not utilise any additional information (such as the content of items) but instead rely purely on user-item interaction data (Breese et al., 1998). Early approaches can be categorised into two memory-based² types: user-based CF and item-based CF. The former predicts a user's rating for a new item by leveraging ratings from similar users who have rated the same item, while the latter estimates the rating by analysing the user's past ratings on similar items (Roy and Dutta, 2022). Taking user-based CF as an example, each user's historical ratings are stored in memory, and similar users are identified and ranked based on these stored ratings during the prediction phase (Wang et al., 2006). The similarity between users, which is leveraged as a weight to calculate the rating, is computed using various heuristic-based approaches, such as the Pearson correlation coefficient (Resnick et al., 1994). Finally, a rating is estimated as a weighted sum of the ratings of the K most similar users for the same item (Adomavicius and Tuzhilin, 2005).

²Some research (Koren, 2008b; Koren et al., 2021) has pointed out that "memory-based" models, which rely on storing and directly referencing past ratings, have evolved to incorporate "model-like" components, such as learning optimised parameters or weights, making them not purely heuristic or memory-driven. Consequently, they suggest that a more precise term for these approaches may be "neighbourhood methods". Nonetheless, since this discussion focuses on the philosophy of early recommendation methods, we retain the traditional terminology for consistency with the original literature.

Providing explanations with memory-based CF is straightforward, as recommendations are directly inferred from users' past behaviours (Hu et al., 2008). The similarity measure in memory-based CF determines the contribution of different "peers" to the final rating prediction. Based on this, the users or items with the highest contributions can be identified as the key explanation behind the recommendation (Hu et al., 2008). For example, Herlocker et al. (2000) presented users with a comparison of how many of their neighbours assigned different rating levels to the recommended item (e.g., 23 neighbours gave high ratings, while only 3 gave low ratings). Fu et al. (2018)³ proposed an item-based CF model, where the items in the user's interaction history that are most similar to the recommended item are used as explanations for the recommendation.

User-based and item-based CF provide different forms of neighbourhood-based explanation. User-based CF can explain a recommendation through the preferences of similar users, whereas item-based CF can relate the recommended item directly to items in the target user's own interaction history. The latter may be easier for users to relate to because the referenced items are already familiar to them, although the usefulness of either explanation form ultimately depends on the user and presentation context. More generally, neighbourhood-based CF explanations remain at the user or item level. They therefore provide less fine-grained information than content-based explanations that can identify specific item characteristics associated with a recommendation (Zhang and Chen, 2020).

1.2.3 Matrix Factorisation (MF)

The model-based CF approach, matrix factorisation, has become widely adopted due to its ability to uncover latent spaces that encode user-item interactions (Wu et al., 2023). In its basic form, user feedback, whether explicit (e.g., numerical ratings) or implicit (e.g., binary interactions), is organised into a matrix, with one dimension representing users and the other representing items (Koren et al., 2009). Through this matrix, MF learns low-dimensional representations of users and items within a shared latent space, where the dimensionality is significantly lower than the typically large number of users or items (Abdollahi and Nasraoui, 2016). Accordingly, each user and

³Theoretically, this method does not fall under memory-based CF, as it maps items into a low-dimensional representation and computes similarity between two items using the inner product. However, since it exhibits characteristics of neighbourhood methods and provides intuitive explanations, we include this study in this discussion.

item is represented by a vector of latent factors (also known as embedding), and their correspondence, i.e., the matching score, is computed using a dot product as a matching function (Deng et al., 2019). As such, MF is regarded as a form of shallow representation learning (Li and Zhao, 2021), as it directly learns fixed-size embeddings from user-item interaction data without incorporating deeper semantic structures or additional information sources. The system optimises the model by fitting the previously observed ratings in the matrix and ultimately uses it to predict future and unknown ratings (Koren et al., 2009).

Beyond this basic framework, several well-known MF variants have been proposed to enhance recommendation performance. Probabilistic Matrix Factorisation (PMF) introduces a probabilistic perspective by assuming Gaussian noise in user-item interactions, thereby improving generalisation (Mnih and Salakhutdinov, 2008). Non-negative Matrix Factorisation (NMF), on the other hand, constrains latent factors to be non-negative, leading to more meaningful representations (Lee and Seung, 1999). Additionally, Singular Value Decomposition++ (SVD++) incorporates implicit feedback to refine rating predictions (Koren, 2008a). These techniques collectively contribute to the robustness and flexibility of MF-based recommendation systems.

Although MF-based approaches substantially improve predictive accuracy over memory-based CF, they often hinder transparent explanations (Deng et al., 2019). This is because users' historical behaviours are abstracted into latent factors, which obscures the direct link between observed behaviours and generated recommendations. In other words, the latent factors in MF models do not inherently possess intuitive meanings (or require manual interpretation), making it difficult to understand why a particular item is recommended over other candidates (Zhang and Chen, 2020). Some studies incorporate additional sources of data, such as tags (Vig et al., 2009) or reviews (Zhang et al., 2014a; Hou et al., 2019), into the MF framework for explanation generation. One pioneering work is that of Zhang et al. (2014a), which extracts item-related aspects from textual reviews along with users' sentiments towards these aspects. This work introduces a new dimension, i.e., aspects, beyond the traditional user-item rating matrix. By jointly considering users' attention to specific aspects and items' performance on these aspects, their approach enables the generation of aspect-level explanations for recommendations.

1.2.4 Neural CF

Neural recommendation models extend conventional CF by using neural networks to learn richer user/item representations, more expressive user-item interaction functions, or both (Wu et al., 2023). These two components can be distinguished conceptually. Representation learning determines how a user and an item are encoded, whereas interaction-function learning determines how the resulting representations are combined to produce a preference score.

Some neural CF models primarily increase the expressiveness of the interaction function. For example, He et al. (2017) used a multilayer perceptron (MLP) to model nonlinear interactions between user and item embeddings, while He et al. (2018a) used a CNN over outer-product interactions. This thesis instead focuses primarily on the representation learning problem: how user and item representations can be constructed from heterogeneous feedback before they are combined for ranking, rating prediction, or explanation.

Different types of representation learning models vary in both input data (e.g., numerical ratings and binary interaction data) and representation modelling techniques (e.g., MLP, AE, and attention mechanisms) (Wu et al., 2023). Both Xue et al. (2017) and Deng et al. (2019) employ MLP to learn deep representations of users and items for item ranking tasks. However, their models differ in terms of input data: the former incorporates both explicit feedback and sampled negative implicit feedback, whereas the latter relies solely on implicit feedback. AE-based models (Sedhain et al., 2015; Wu et al., 2016; Liang et al., 2018) generally consist of two key components: an encoder, which maps the input data (i.e., each user's or item's historical interactions) into a hidden representation; and a decoder, which reconstructs the input data to predict missing interactions. The input data for these models can be either explicit (Sedhain et al., 2015) or implicit (Wu et al., 2016; Liang et al., 2018) feedback.

Attention mechanisms provide another way to assign different importance to the information used to construct user and item representations. Chen et al. (2017), for example, assigned different weights to a user's historical items according to their importance to that user when constructing the user representation. In contrast, He et al. (2018b) and Lv et al. (2020) made the weighting target-aware by considering the relevance of each historical item to the current candidate item. As a result, the representation of the same user could vary across recommendation targets. Analogous aggregation can be applied on the item side by constructing an item representation

from information associated with its neighbouring users. These approaches illustrate an important departure from fixed user and item representations: the information emphasised when constructing a representation can vary with the current user-item interaction.

Because these models assign importance to historical interactions, their attention weights or highly weighted historical items can also provide information about which previous interactions contributed to the recommendation. Such signals can support model interpretation, although attention weights should not automatically be treated as evidence of user-perceived explanation quality.

1.2.5 Graph Neural Network (GNN)-Based Methods

GNN-based recommendation methods represent users, items, and potentially other entities as nodes and their relationships as edges. A typical GNN layer performs message propagation by obtaining information from neighbouring nodes, aggregation by combining the incoming messages, and update by using the aggregated information to construct the node representation for the next layer. Repeating this process across multiple layers allows user and item representations to incorporate information from higher-order collaborative neighbourhoods (Wang et al., 2019; Wu et al., 2023). Different GNN variants mainly differ in how messages are transformed, aggregated, and weighted. GCNs (Kipf and Welling, 2016), for example, use graph convolution operations, whereas GATs (Velickovic et al., 2018) learn attention weights for neighbouring nodes.

Wang et al. (2019) applied graph convolutional operations to the user-item interaction graph, refining node representations through three key propagation steps: feature transformation, neighbourhood aggregation, and non-linear activation. However, subsequent work has questioned the necessity of such complex operations in CF settings, where only user-item interaction data are available. In particular, feature transformations and non-linear activations may introduce unnecessary complexity without yielding commensurate gains in performance. To address this concern, simplified neural graph CF models have been proposed. For instance, Chen et al. (2020a) and He et al. (2020) developed lightweight GCN variants that omit redundant deep learning components while still achieving strong empirical performance.

Despite their strong representation-learning capability, standard GNN-based recommenders do not inherently provide user-facing explanations because their predictions are usually produced from latent node embeddings (Ma et al., 2023). One approach is to augment the user-item graph with external semantic structure. For example, Ma et al. (2023) integrated user-item interactions with a knowledge graph containing external entities and relations associated with items. It then extracted and weighted multi-hop reasoning paths between users and candidate items, allowing selected paths to be presented as evidence for a recommendation. Another approach is to connect graph representations with textual information. Shuai et al. (2023) introduced topic nodes and review information into a GNN and selected review sentences that were most indicative of the learned user-item interaction. These studies illustrate that GNN representations can support explanation when the graph is augmented with identifiable semantic entities, paths, topics, or textual evidence rather than relying only on latent user and item embeddings.

1.2.6 Review-Based Methods

To enrich user and item representations beyond interaction histories, review-based recommender systems incorporate information extracted from textual reviews. Reviews can provide fine-grained descriptions of users' preferences and item characteristics that are not directly represented by overall ratings or binary interactions (McAuley and Leskovec, 2013). Recent surveys show that review-based recommendation encompasses diverse approaches to extracting and modelling information from reviews, ranging from aspect and sentiment analysis to deep neural and language-model-based representation learning (Hasan et al., 2025). According to the level at which review information is represented in this thesis, we organise representative methods into three broad groups: topic-level, review-level, and aspect-level approaches. The following discussion focuses on how each group constructs user/item representations and, where applicable, how those representations support recommendation explanations.

Topic-Level Models

Topic-level models⁴ treat all reviews of a user/item, each individual review, or even each sentence as a document to extract a k -dimensional topic distribution (i.e., a stochastic vector), where each value represents the probability that the document pertains to the corresponding topic. Meanwhile, each topic corresponds to a word distribution (McAuley and Leskovec, 2013). Based on this, users or items can be represented through topic distributions. For example, McAuley and Leskovec (2013) linked latent factors learned from user and item identifiers, referred to in this thesis as ID embeddings⁵, with topic distributions learned from reviews, and thus set the number of latent factors equal to the number of review topics. They transformed each latent factor dimension into its corresponding topic distribution dimension using a scaling parameter, while ensuring the topic distribution remained normalised.

Hou et al. (2019) constructed a user-topic preference matrix and an item-topic quality matrix from user and item reviews, respectively. The former captures each user's attention to various topics, and the latter reflects sentiment expressed in reviews about item-related topics. They assumed that latent factors learned from the rating matrix and those decomposed from the topic matrices are shared for both users and items. In contrast, Sun et al. (2021) did not assume any shared structure between topic distributions and latent factors; instead, they concatenated user and item topic distributions as well as their latent factors, passed them through MLPs, and summed the outputs to obtain the final rating prediction.

For explanation generation, topic-level models typically rely on the match between user and item topic distributions, or on top words from the top-ranked topics. Although McAuley and Leskovec (2013) used topic information to interpret the semantic meaning of latent factors, the explanation was primarily attached to individual user or item factors rather than constructed specifically for a particular user-item recommendation pair. It therefore helped interpret what latent dimensions represented, but did not directly explain why a particular item was recommended to a particular user.

⁴Some topic-level methods refer to their latent topics as "aspects". In this thesis, the distinction is based on the type of representation produced: topic-level methods infer latent topic distributions from review text, whereas aspect-level methods represent identifiable item characteristics mentioned or implied in reviews, often together with associated opinions or sentiment. We therefore categorise methods based on latent topic distributions as topic-level even when the original authors use the term "aspect".

⁵An ID embedding is a trainable vector representation associated with a discrete user or item identifier, typically implemented through an embedding lookup table.

Hou et al. (2019) presented a radar chart as a visual explanation, where each axis corresponds to a topic (e.g., room and location), and both user preferences and item qualities are plotted. The similarity in shape between the two profiles serves as an intuitive justification for the recommendation. Unlike most topic-level models, which provides explanations in the form of topics or keywords, Sun et al. (2021) generated review-like explanatory sentences through a Gated Recurrent Unit (GRU) structure, making its explanation mechanism closer to review-level methods.

Although topic-level models can improve transparency, their explanations rely on manually assigned topic names, which are typically derived by summarising the keywords associated with each topic. This process is inherently subjective and may vary across annotators. Moreover, the number of topics must be carefully controlled to maintain explainability. A large number of topics makes manual annotation costly and impractical, whereas too few topics may lead to overly coarse representations, resulting in semantic overlap and limited discriminative power between topics. As a consequence, the quality and granularity of explanations provided by topic-level models are highly sensitive to the choice of topic number. To produce explanations that are less dependent on manual annotation and more closely aligned with how users express their own preferences in natural language, some studies have explored alternative ways of leveraging reviews, such as review-level and aspect-level methods.

Review-Level Models

In review-level models, each review is typically represented as a sequence of word embeddings. These embeddings can be obtained in several ways: (1) learning a task-specific embedding matrix during training from one-hot inputs (Seo et al., 2017); (2) pretraining word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) embeddings on the in-domain corpus (Catherine and Cohen, 2017); or (3) directly adopting fixed embeddings pretrained on large external corpora (e.g., Google News (Mikolov et al., 2013)) (Zheng et al., 2017). The first approach learns embeddings jointly with the recommendation task, whereas the latter two typically use fixed, non-trainable embeddings.

After obtaining word embeddings, a user's or an item's historical reviews are either concatenated into a single document and processed jointly, or encoded individually and subsequently aggregated to form review representations. For example, DeepCoNN (Zheng et al., 2017) constructs user and item representations solely from textual reviews, without relying on ID embeddings. It concatenates all reviews for each user

and item into a single document, represents the document as a word embedding tensor, and applies a CNN to extract high-level semantic features. The resulting user and item representations are then combined through a shared layer for rating prediction. However, Catherine and Cohen (2017) observed that DeepCoNN's strong performance largely arises from exploiting the target review, which is unavailable at test time. To address this issue, they proposed TransNets, which approximates the latent representation of the target review via a CNN-based transformation network. An extended variant, TransNet-Ext, further incorporates ID embeddings to improve predictive performance.

Similarly, D-Attn (Seo et al., 2017) derives user and item representations solely from reviews using a dual attention mechanism, with local attention emphasising key phrases and global attention capturing overall semantics, before feeding the resulting representations into CNNs for rating prediction. NARRE (Chen et al., 2018) encodes each review using a CNN and aggregates them via an attention mechanism that assigns higher weights to more informative reviews. It further incorporates ID embeddings, which are combined with review embeddings for rating prediction. DualPC (Sun et al., 2020) adopts a dual-task framework, in which one task predicts ratings using ID embeddings and review embeddings, while the other reconstructs personalised reviews from ID embeddings and ratings through a Long Short-Term Memory (LSTM), thereby enforcing consistency between rating prediction and review generation.

These models benefit from the ability of neural networks to extract deep semantic representations from textual reviews, which contributes to improved recommendation accuracy. However, they also risk introducing irrelevant or redundant information, as full-length reviews often contain details that are not essential for effective recommendation.

In terms of explainability, some review-level models generate predicted reviews or highlight existing reviews that are most indicative of the recommendation outcome. DualPC (Sun et al., 2020) provides personalised explanations by generating review text for the recommended item, guided by the alignment between rating prediction and review generation. TransNets (Catherine and Cohen, 2017), although not explicitly designed for explanation, identifies existing item reviews that are most similar to the predicted latent representation of the target review. Likewise, NARRE (Chen et al., 2018) leverages attention weights to highlight the most informative reviews associated with the recommended item, thereby helping users understand the recommendation rationale. D-Attn (Seo et al., 2017) offers word-level interpretability by visualising local

and global attention distributions; however, it does not produce explicit explanations for unseen items at recommendation time, as its attention mechanisms primarily serve to interpret rating predictions rather than to justify recommendations.

Compared with topic-level explanations, which typically rely on keywords or manually summarised topics, review-level explanations provide complete, human-readable sentences in natural language, making them easier for users to understand and potentially more persuasive. Nevertheless, presenting full reviews, whether generated or retrieved, may not always be practical or user-friendly, particularly when reviews are long or contain redundant information. As a result, review-level explanations can become overwhelming or insufficiently focused in certain scenarios.

Aspect-Level Models

Aspect-level methods transform textual reviews into structured information about specific item characteristics and users' evaluations of those characteristics. This requires an additional preprocessing stage, commonly framed as **Aspect-Based Sentiment Analysis (ABSA)**. ABSA identifies aspects and their associated opinions or sentiment polarities, and aspects may either be explicitly expressed in the text or inferred from context (Hua et al., 2024). Earlier recommendation studies commonly relied on specialised sentiment-analysis pipelines such as Sentires⁶, while more recent ABSA research has also investigated transformer-based and generative language-model approaches (Hua et al., 2024). In this thesis, the structured information consists of an aspect and its associated sentiment polarity. After extraction, these signals can be aggregated or encoded to form fine-grained user and item representations.

A common formulation aggregates aspect-level information into a **user-aspect attention matrix** and an **item-aspect quality matrix**. The former quantifies how strongly each user attends to individual aspects, typically using the frequency with which those aspects are mentioned in the user's reviews. The latter quantifies how an item is evaluated on each aspect by combining aspect frequency with the sentiment expressed towards that aspect. EFM (Zhang et al., 2014a) factorised these matrices to learn explicit user, item, and aspect factors. The resulting explicit user and item factors were then combined with latent factors learned from user/item IDs within a unified matrix-factorisation framework for rating prediction. This design preserved two complementary sources of information: fine-grained, human-interpretable aspect information was

⁶<https://github.com/evison/Sentires>

modelled through the explicit factors, while latent collaborative information was captured separately through ID embeddings before the two were combined for rating prediction.

MTER (Wang et al., 2018) extended aspect-sentiment modelling by additionally incorporating opinion phrases extracted from reviews. It constructed three 3-dimensional tensors: a user-item-aspect tensor capturing aspect-level user preferences over items, a user-aspect-opinion tensor, and an item-aspect-opinion tensor. Joint tensor factorisation was then used to learn shared representations of users, items, aspects, and opinion phrases. This enabled recommendation and explanation to be learned within the same multi-task framework and allowed explanations to contain more detailed aspect-opinion information rather than aspects alone.

Some aspect-level models further incorporated semantic relationships among aspects to reduce the effect of sparse or overlapping aspect expressions. DEAML (Gao et al., 2019), for example, organised aspects into a hierarchical semantic structure, where fine-grained concepts could be grouped under broader parent concepts. User-aspect attention and item-aspect quality representations were constructed at different levels of this hierarchy following the general formulation of EFM (Zhang et al., 2014a). The representations learned at different semantic levels were then integrated through an attentive multi-view framework. This allowed the model to combine fine-grained and broader semantic information rather than treating every extracted aspect as an unrelated feature.

AARM (Guan et al., 2019) also incorporated semantic aspect representations, but it focused on aspect information without explicitly modelling the corresponding opinion terms or sentiment polarities. To alleviate the sparsity caused by exact aspect matching, AARM modelled interactions between all aspects associated with a user and all aspects associated with a target item, allowing semantically related aspects to interact even when their surface forms differed. Attention mechanisms were used to emphasise informative aspect interactions and to construct preference information conditioned on the target item. Consequently, the aspects emphasised for the same user could vary across candidate items.

A2CF (Chen et al., 2020b) replaced the linear matrix-factorisation formulation with residual feed-forward networks to model the observed values in the user-aspect attention and item-aspect quality matrices. In addition to learning user and item representations from these matrices, A2CF explicitly incorporated aspect representations into preference prediction. Aspect-level contributions were weighted according to the

user’s modelled attention to each aspect and the item’s modelled quality on that aspect, allowing the final prediction to reflect fine-grained compatibility between user preferences and item characteristics.

HDE (Luo et al., 2019) combined review-level and aspect-level information. User and item reviews were encoded into semantic review representations, while decoder networks were trained to reconstruct the corresponding user-aspect attention and item-aspect quality vectors. The aspect-level vectors therefore acted as supervisory signals for learning review representations. HDE also incorporated collaborative neighbourhood information by aggregating ID embeddings of interacted items or users through attention. Its final user and item representations thus combined review semantics, aspect-level supervision, and neighbourhood information for rating prediction.

In terms of explanation, aspect-level models naturally yield more targeted justifications by explicitly linking users’ aspect-level preferences to an item’s performance. EFM (Zhang et al., 2014a) identifies the aspects a user values (via the learned user-aspect attention matrix) and assesses the item’s performance on those aspects (via the learned item-aspect quality matrix). For a recommended item, it selects the highest-performing aspect among those valued by the user as the explanation; for a disrecommended item, it selects the lowest-performing one. A2CF (Chen et al., 2020b) also generates personalised explanations based on the learned attention and quality matrices by aligning the user’s preferences with the aspect-level differences between the recommended substitute and the original item; the top-ranked aspects are then used to compose the final explanation. HDE (Luo et al., 2019) determines the top aspects a user values and those on which an item performs well, using the learned attention and quality vectors. The intersection of these sets is then used to produce aspect-level explanations. Additionally, HDE visualises user and item profiles as aspect clouds, allowing users to intuitively assess how well an item matches their preferences. MTER (Wang et al., 2018) first identifies the top aspects a user values for a recommended item, based on predicted user-item-aspect affinities. For each selected aspect, it retrieves the top opinion phrases by combining user-aspect-opinion and item-aspect-opinion affinities. These aspect-opinion pairs form the basis for fine-grained, personalised explanations. DEAML (Gao et al., 2019) explains recommendations by identifying matched aspects across different semantic levels in the aspect hierarchy, highlighting those that the user values and the item performs well on, based on attention-guided representations. Although AARM (Guan et al., 2019) does not generate explicit recommendation explanations, it enhances interpretability through attention mechanisms that highlight

influential aspect interactions. These can be analysed to understand the reasoning behind the model's outputs.

Aspect-based explanations provide a fine-grained way to relate a recommendation to identifiable item characteristics. Unlike full-review explanations, they can present a small number of structured attributes, and unlike latent-factor explanations, the selected aspects have direct semantic meanings. Their usefulness depends on whether the predicted aspects accurately reflect the user's preferences and the item's characteristics and, ultimately, on how users perceive the resulting explanation. This distinction between automatic aspect-prediction quality and user-perceived explanation quality is important for the evaluation adopted later in this thesis.

Structured aspect-level information also appears in practical e-commerce interfaces. For example, Tmall has employed a review-tag function that summarises frequently mentioned product features together with their associated evaluations and frequencies, allowing consumers to inspect recurring opinions without reading all individual reviews (Bao et al., 2021). Although these review tags summarise collective product feedback rather than explain a personalised recommendation, they illustrate how concise, structured information about product characteristics can support users in evaluating items.

1.3 Research Problem and Challenges

This thesis addresses how to learn user and item representations from heterogeneous feedback to support accurate and explainable recommendation. Two related issues arise: preserving the distinct information carried by different feedback types and allowing representations to vary with the user-item interaction being modelled. Implicit interactions record users' behaviour towards items, explicit ratings provide evaluative preference information, and textual reviews provide fine-grained reasons for those evaluations. Modern representation learning methods can combine these sources of information, but combining them alone does not determine how their different semantics should be preserved or how the most relevant information should be reflected in a particular interaction. This thesis therefore focuses on two related representation learning challenges.

C1 Preserving and connecting feedback-specific representations when learning from implicit and explicit feedback.

Implicit interactions and explicit ratings provide complementary but different information. Earlier unified recommendation methods demonstrated that combining them could improve recommendation performance (Liu et al., 2010; Jadidinejad et al., 2019), but existing approaches differ substantially in whether the two feedback types share representations, are transformed into one another, or remain separately represented. Recent multi-behaviour recommendation research has similarly shown the importance of preserving behaviour-specific structures while modelling dependencies across feedback types (Zhao et al., 2024a; Kim et al., 2026). Much of this literature, however, focuses on multiple implicit behaviours such as views, clicks, add-to-cart actions, and purchases, often using auxiliary behaviours to improve prediction of a designated target behaviour.

The problem addressed in this thesis is more specific: how to maintain distinct representations for implicit behavioural information and explicit evaluative information while allowing useful information to transfer between them, and how to use the resulting representations for two corresponding objectives, item ranking and rating prediction. This requires a representation mechanism that preserves feedback-specific semantics while modelling their interdependence.

C2 Learning interaction-dependent user and item representations for accurate and aspect-level explainable recommendation.

A second problem concerns the assumption that one fixed representation of a user or item is appropriate across all interaction counterparts. Target-aware recommendation research has shown that different parts of a user’s history can become relevant for different candidate items (He et al., 2018b; Cao et al., 2023; Kim et al., 2024), and broader context-aware recommendation research continues to investigate how contextual information should be represented and incorporated into recommendation models (Mateos and Bellogín, 2024). These developments mean that interaction-dependent representation itself should not be treated as a new general concept.

The specific challenge investigated in this thesis is how to construct both an item-dependent user representation and a user-dependent item representation in an aspect-level recommendation setting. Numerical ratings, textual reviews, and collaborative neighbourhoods provide different information about an interaction. The representation learning problem is therefore how to combine general collaborative information with fine-grained aspect-sentiment information so that

the representation of each entity reflects its current counterpart and can support both rating prediction and positive-aspect prediction.

1.4 Research Questions

The research problem identified in Section 1.3 concerns how user and item representations can preserve information that varies across feedback types and across interaction counterparts. This leads to the overarching research question of this thesis: **How can user and item representations be learned from heterogeneous feedback to account for differences across feedback types and user-item interactions while supporting accurate and aspect-level explainable recommendation?**

This overarching question was investigated through two research questions corresponding to the two challenges identified above:

RQ1 How can implicit and explicit feedback be integrated to learn user and item representations that preserve feedback-specific information and support both item ranking and rating prediction?

RQ1 addressed Challenge C1 by investigating how feedback-specific user and item representations could be preserved while allowing information to be exchanged through collaborative neighbourhoods. Chapter 3 addressed this question through Interdependent Embedding Learning (IEL), which constructed separate implicit and explicit feedback graphs and connected their feedback-specific representations through intra-graph and inter-graph message passing.

RQ2 How can user and item representations be conditioned on each other to support both rating prediction and aspect-level recommendation explanation?

RQ2 addressed Challenge C2 by investigating how user and item representations could be conditioned on each other for a target interaction. Chapter 4 addressed this question through Dynamic⁷ Representation Learning (DRL), which combined general collaborative representations with aspect-sentiment information to construct item-dependent user representations and user-dependent item representations for rating and positive-aspect prediction.

⁷Throughout this thesis, the term "dynamic" refers to interaction-dependent meaning rather than temporal preference evolution.

1.5 Research Methodology

This thesis investigated the two research questions through two empirical modelling studies organised under a common **polymorphic representation learning** perspective. In this thesis, polymorphic representation learning refers to the idea that the same user or item need not be represented by one invariant vector. Instead, its representation may differ according to the feedback type being modelled or the interaction counterpart for which a prediction is made. This perspective provides a common interpretation of the two studies rather than constituting a separate recommendation algorithm.

1. Interdependent Embedding Learning (IEL): feedback-dependent representations.

Chapter 3 investigated RQ1 using IEL. Separate user and item representations were learned for implicit and explicit feedback. Two corresponding user-item graphs were constructed, and GCN-based message passing was performed both within the feedback-specific graphs and through selected cross-feedback pathways. This allowed the implicit and explicit representations to remain distinct while exchanging collaborative information. IEL was evaluated for both item ranking and rating prediction.

2. Dynamic Representation Learning (DRL): interaction-dependent representations.

Chapter 4 investigated RQ2 using DRL. A rating-based GNN first learned general collaborative representations of users and items. For a target user-item pair, the dynamic representation layer used the other entity's general representation to attend to the entity's own general representation and review-derived aspect-sentiment information, producing interaction-dependent user and item representations. The resulting representations were evaluated for rating prediction and positive-aspect prediction.

For the DRL study, **a prompt-based LLM procedure was employed to construct the aspect-sentiment data from textual reviews**. This procedure was adopted following a preliminary analysis using Sentires, which revealed practical limitations for the downstream aspect-level modelling setting considered in this study. The specific model and API implementation are described in Chapter 4.

Both studies were evaluated through offline experiments on public Amazon review datasets. Recommendation performance was measured using task-appropriate

metrics, including NDCG for ranking and MSE/MAE for rating prediction. DRL’s aspect-level outputs were evaluated using $\text{Hit}@N$, $\text{Recall}@N$ and $\text{Precision}@N$, and ablation studies were conducted to examine the contribution of individual architectural components. These aspect-prediction metrics provided an operational evaluation of explanation rather than a direct measurement of user-perceived explanation quality.

1.6 Research Contributions

The main contributions of this thesis are as follows:

1. **A unifying polymorphic representation perspective.**

This thesis formulated two forms of representation variation within a common perspective: feedback-dependent representation, in which the same user or item was represented differently under implicit and explicit feedback, and interaction-dependent representation, in which the representation of a user or item depended on the current interacting counterpart. This perspective connected the two empirical studies without assuming that heterogeneous feedback or interaction contexts should be compressed into a single invariant representation.

2. **Interdependent representation learning from implicit and explicit feedback.**

IEL maintained separate user and item representations for implicit and explicit feedback and connected selected feedback-specific representations through GCN-based intra-graph and inter-graph message passing. This study investigated how collaborative neighbourhood information could connect behavioural and evaluative representations while jointly supporting item ranking and rating prediction. The experimental results showed improvements relative to the compared unified recommendation baselines, with substantially stronger benefits for ranking.

3. **Interaction-dependent representation learning for aspect-level recommendation.**

DRL combined general collaborative representations with aspect-sentiment information to construct an item-dependent user representation and a user-dependent item representation. The model improved rating prediction and positive-aspect prediction relative to the compared baselines across the evaluated datasets. The ablation analysis also identified limitations of individual components, including

the inconsistent contribution of semantic aspect embeddings across prediction objectives.

4. LLM-processed aspect-sentiment data for recommendation research.

This thesis constructed three derived Amazon review datasets containing structured aspect-sentiment information produced through a prompt-based LLM extraction procedure. These datasets supported the aspect-level representation learning and evaluation conducted in Chapter 4 and provide a resource for further research on aspect-level and explainable recommendation.

1.7 Thesis Outline

The remainder of this thesis is structured as follows:

- **Chapter 2: Literature Review.** Reviews the foundations of recommender systems, representation-learning methods, learning from heterogeneous feedback, review-based recommendation, and explainable recommendation.
- **Chapter 3: Interdependent Embedding Learning (IEL).** Investigates RQ1 and presents the GCN-based framework for learning distinct yet interdependent representations from implicit and explicit feedback for ranking and rating prediction.
- **Chapter 4: Dynamic Representation Learning (DRL).** Investigates RQ2 and presents the interaction-dependent aspect-level recommendation framework, including the construction of aspect-sentiment data, the rating-based GNN, the dynamic representation mechanism, and the evaluation of rating and positive-aspect prediction.
- **Chapter 5: Discussion.** Synthesises the findings of IEL and DRL, relates them to the overarching polymorphic representation perspective, discusses the significance and broader impact of the thesis, and identifies limitations and directions for future research.
- **Chapter 6: Conclusions.** Summarises the principal findings and contributions of the thesis.

Overall, this thesis investigated how user and item representations could preserve information that varies across feedback types and interaction counterparts. Through IEL and DRL, it examined feedback-dependent and interaction-dependent representation learning as two complementary approaches to modelling heterogeneous feedback for accurate and aspect-level explainable recommendation.

Chapter 2

Literature Review

Modern collaborative filtering (CF) research can be viewed as addressing two closely related problems: how users and items are represented and how the resulting representations are combined to estimate preference (Wu et al., 2023). **Representation learning** constructs representations that encode information about user preferences and item characteristics from interaction data and, where available, auxiliary information. **Interaction modelling** then determines how these representations are combined to produce ranking or rating predictions. This thesis focuses primarily on the representation-learning problem, particularly when the available feedback contains heterogeneous or interaction-dependent information.

Predictive accuracy has historically been a major objective of recommender systems, but explainability has also become an important research concern (Zhang and Chen, 2020). Explanations may serve different purposes rather than representing a single measurable property. Tintarev (2007), for example, identified seven potential aims: *transparency, scrutability, trustworthiness, effectiveness, persuasiveness, efficiency, and satisfaction*. These aims describe different ways in which explanations may support users and system designers, but their effects depend on the explanation design, user characteristics, and evaluation setting (Wardatzky et al., 2025). Table 2.1 summarises these aims.

Textual reviews provide an additional source of information for user and item representation learning because they contain descriptions of item characteristics, user opinions, and the reasons associated with overall evaluations. Review-based recommender systems incorporate such textual information in different ways, including learning representations from complete reviews, extracting latent topics, and modelling fine-grained aspects and sentiment (Hasan et al., 2025). These representations can complement ratings or binary interactions and can also provide human-readable information that may be used for recommendation explanations.

TABLE 2.1: Aims of explanations in recommender systems (adapted from Tintarev (2007)).

Aim	Definition
Transparency	Explain how the system works
Scrutability	Allow users to tell the system it is wrong
Trustworthiness	Increase users' confidence in the system
Effectiveness	Help users make good decisions
Persuasiveness	Convince users to try or buy
Efficiency	Help users make decisions faster
Satisfaction	Increase the ease of usability or enjoyment

Among review-based approaches, aspect-level models focus on specific item characteristics and users' evaluations of those characteristics (Zhang et al., 2014a; Gao et al., 2019; Chen et al., 2020b). In Aspect-Based Sentiment Analysis (ABSA), aspects may be explicitly expressed as words or phrases in the review or inferred from the surrounding context (Hua et al., 2024). Aspect and sentiment information can subsequently be aggregated into fine-grained user and item representations. Because the resulting features have identifiable semantic meanings, they can also provide structured evidence for aspect-level explanations, although the quality of such explanations requires separate evaluation.

Accordingly, this chapter reviews the literature along the main dimensions relevant to the research questions of this thesis. Section 2.1 examines multiple representation learning, first across multiple feedback types and then within a single-feedback setting. Section 2.2 reviews target-aware and interaction-dependent representation learning, focusing on how representations can vary according to the current interaction counterpart. Section 2.3 then reviews aspect-level recommendation models, followed by explanation methods and forms in Section 2.4 and approaches for evaluating recommendation explanations in Section 2.5. Mathematical formulations are included selectively where they help make explicit how representative methods construct user and item representations, exchange information across feedback types, or operationalise explanation evaluation; formulations that are not directly relevant to these comparisons are omitted.

2.1 Multiple Representation Learning

Some recommender systems learn more than one representation for the same user or item. Multiple representations can arise because different feedback types convey different information about the entity, or because the same entity plays different roles within a prediction process. This section first reviews models that maintain multiple representations across feedback types and then considers models that learn multiple representations within a single feedback setting.

2.1.1 Multiple Feedback

Koren (2008a) proposed a classic latent factor model, **SVD++** (Singular Value Decomposition++), which exploits both explicit and implicit feedback for rating prediction. In SVD++, each explicit user embedding $\mathbf{p}$ is augmented by the averaged embeddings of items the user has implicitly interacted with, thereby incorporating both explicit and implicit signals into the user representation. Importantly, distinct item embeddings are learned for different feedback types: each item is associated with two embeddings, $\mathbf{q}$ and $\mathbf{y}$, for explicit and implicit feedback, respectively. The prediction rule is as follows¹:

$$\hat{r}_{ui} = \mathbf{q}_i^\top \left(\mathbf{p}_u + \frac{1}{\sqrt{|\mathcal{I}_u|}} \sum_{j \in \mathcal{I}_u} \mathbf{y}_j \right), \quad (2.1)$$

where $\mathcal{I}_u$ is the set of items with which user u has interacted implicitly. Compared to traditional SVD (Paterek, 2007), which does not incorporate the implicit feedback term in Equation 2.1 (i.e., $\hat{r}_{ui} = \mathbf{q}_i^\top \mathbf{p}_u$), SVD++ achieved significantly higher accuracy in rating prediction, largely due to the inclusion of implicit feedback.

Notably, Koren (2008a) also proposed an Asymmetric-SVD model:

$$\hat{r}_{ui} = \mathbf{q}_i^\top \left(\frac{1}{\sqrt{|\mathcal{R}_u|}} \sum_{j \in \mathcal{R}_u} r_{uj} \mathbf{x}_j + \frac{1}{\sqrt{|\mathcal{I}_u|}} \sum_{j \in \mathcal{I}_u} \mathbf{y}_j \right), \quad (2.2)$$

where $\mathcal{R}_u$ denotes the set of items rated by user u , and $\mathbf{x}_j$ represents item j 's embedding for modelling user representation based on explicit feedback, weighted by rating r_{uj} . That is, the explicit user embedding $\mathbf{p}$ is replaced by the aggregation of the embeddings of their previously rated items. However, experiments showed that SVD++

¹For clarity, we omit the bias terms (e.g., user bias b_u , item bias b_i , and the global bias μ) from the original formulation, as well as in some of the subsequent models.

outperformed Asymmetric-SVD, suggesting the value of having a free user embedding in the rating prediction task.

Chen and Peng (2018) proposed a model referred to as **PMF+P+N** (Probabilistic Matrix Factorisation with Positive and Negative implicit feedback), where each item is associated with two embeddings ($\mathbf{z}$ and $\mathbf{v}$) learned from implicit and explicit feedback, respectively. The user embedding $\mathbf{u}$ is shared across both types of feedback. Prediction is based on the inner product between user and item embeddings: $\mathbf{u}^\top \mathbf{z}$ for implicit feedback and $\mathbf{u}^\top \mathbf{v}$ for explicit feedback. PMF+P+N supports dual-task learning and defines the loss function² as a combination of a pairwise log-likelihood loss (for ranking) and a squared error (for rating), controlled by a weight parameter³ α :

$$\begin{aligned} \mathcal{L} = & -\alpha \left(\sum_{(i,k) \in \mathcal{P}} \ln \sigma(\mathbf{u}_i^\top \mathbf{z}_k) + \sum_{(i,k) \in \mathcal{N}} \ln (1 - w \cdot \sigma(\mathbf{u}_i^\top \mathbf{z}_k)) \right) \\ & + (1 - \alpha) \sum_{(i,j) \in \mathcal{R}} \left(\sigma(\mathbf{u}_i^\top \mathbf{v}_j) - r_{ij} \right)^2. \end{aligned} \quad (2.3)$$

Here, $\mathcal{P}$ represents the set of observed implicit interactions, while $\mathcal{N}$ denotes the set of negative instances, which may either include all or be sampled from the unobserved interactions. The hyperparameter $w \in [0, 1]$ is used to scale the relative importance of the negative samples. The sigmoid function $\sigma(x)$ is used to bound predictions:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \quad (2.4)$$

Chen and Peng (2018) reported that PMF+P+N outperformed the compared single-task models in both ranking and rating prediction. The reported experiments, however, did not isolate the contribution of the separate item embeddings from that of joint task optimisation, because no ablation retained the feedback-specific item embeddings while removing one of the two learning objectives. The highest rating prediction accuracy was obtained when α was between 0 and 1, providing evidence that incorporating the ranking objective benefited rating prediction under this model. The authors did not report a corresponding analysis of how varying α affected ranking performance. They also reported that their negative-sampling strategy did not improve rating prediction.

²For ease of exposition, regularisation terms are omitted here and in the subsequent loss functions.

³To present in a clearer and more concise manner, we slightly reformulate the expression of task weighting in the loss function. This reformulation remains semantically equivalent to the original formulation.

Mandal and Maiti (2020) proposed the **GPMF** (Generalised Probabilistic Matrix Factorisation) model, which leverages multiple explicit and implicit feedback signals to enhance user embedding quality. In GPMF, item embeddings are learned separately under different types of feedback. While distinct user embeddings are also learned for each feedback type, these embeddings are regularised to remain close to the user embedding learned from explicit ratings, which acts as an anchor. Formally, let $\mathcal{R}$ denote the set of rating interactions, and $\mathcal{D}_x$ represent the set of additional feedback (e.g., thumbs up/down or clicks), where $x \in \{1, \dots, l\}$ indexes the l alternative feedback types. The overall loss function of GPMF is defined as:

$$\begin{aligned} \mathcal{L} = & \sum_{(i,j) \in \mathcal{R}} \left(\sigma \left(\mathbf{u}_i^\top \mathbf{v}_j \right) - r_{ij} \right)^2 \\ & + \sum_{x=1}^l \left(\lambda_x \sum_{(i,j) \in \mathcal{D}_x} \left(\sigma \left(\mathbf{p}_{x,i}^\top \mathbf{q}_{x,j} \right) - d_{x,ij} \right)^2 + \lambda_{u,x} \sum_{i=1}^m \left\| \mathbf{u}_i - \mathbf{p}_{x,i} \right\|_2^2 \right), \end{aligned} \quad (2.5)$$

where $\mathbf{p}_{x,i}$ and $\mathbf{q}_{x,j}$ are the user and item embeddings learned from the x -th feedback type $\mathcal{D}_x$, respectively, and $\mathbf{u}_i$ is the user embedding from the rating data. The squared L2 norm term enforces proximity between $\mathbf{p}_{x,i}$ and $\mathbf{u}_i$. Hyperparameters λ_x and $\lambda_{u,x}$ control the relative importance of the reconstruction and regularisation terms for each feedback type.

In Mandal and Maiti (2020)'s work, they used the predicted ratings $\sigma \left(\mathbf{u}_i^\top \mathbf{v}_j \right)$ to evaluate both rating prediction and ranking performance. They instantiated GPMF with one type of explicit feedback (a reliability score⁴) and one type of implicit feedback (views), and referred to this variant as PMFEI. The corresponding loss function is defined as:

$$\begin{aligned} \mathcal{L}_{EI} = & \sum_{(i,j) \in \mathcal{R}} \left(\sigma \left(\mathbf{u}_i^\top \mathbf{v}_j \right) - r_{ij} \right)^2 \\ & + \lambda_E \sum_{(i,j) \in \mathcal{D}_E} \left(\sigma \left(\mathbf{p}_{E,i}^\top \mathbf{q}_{E,j} \right) - d_{E,ij} \right)^2 + \lambda_{u,E} \sum_{i=1}^m \left\| \mathbf{u}_i - \mathbf{p}_{E,i} \right\|_2^2 \\ & + \lambda_I \sum_{(i,j) \in \mathcal{D}_I} \left(\sigma \left(\mathbf{p}_{I,i}^\top \mathbf{q}_{I,j} \right) - d_{I,ij} \right)^2 + \lambda_{u,I} \sum_{i=1}^m \left\| \mathbf{u}_i - \mathbf{p}_{I,i} \right\|_2^2. \end{aligned} \quad (2.6)$$

Mandal and Maiti (2020) introduced two simplified variants: PMFE and PMFI, which

⁴Interested readers are referred to the original paper for more details.

include only explicit and implicit feedback, respectively, by removing the corresponding terms from Equation 2.6. They reported that PMFEI outperformed both PMFE and PMFI on the evaluated rating prediction and ranking tasks. Within their experimental setting, PMFE also outperformed PMFI, indicating that the explicit auxiliary signal used in PMFE provided more useful information than the implicit view signal used in PMFI for these datasets. PMFEI also outperformed PMF+P+N (Chen and Peng, 2018).

From a modelling perspective, GPMF learned distinct user and item embeddings for different feedback types while regularising the user embeddings towards the representation learned from ratings. This introduced a soft relationship between the feedback-specific representation spaces. Compared with PMF+P+N, the design allowed greater representational flexibility, although it also introduced additional feedback-specific parameters.

Hadash et al. (2018) also introduced a dual-task framework named **RnR** (Rank and Rate), which simulates a user's two-phase decision-making: first deciding whether to interact with an item (ranking) and then rating it (rating prediction). RnR assigns different embeddings to users and items across these phases. Specifically, the item embedding q_i is adjusted by a deviation vector q_i^d , and both the user and item embeddings are passed through a fully connected layer with a non-linear activation function to produce post-consumption embeddings p_u^P and q_i^P . The total loss is defined as:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{rank}}(f_{\text{rank}}, \mathcal{R}; p_u, q_i) + (1 - \alpha) \cdot \mathcal{L}_{\text{rating}}(f_{\text{rating}}, \mathcal{R}; p_u^P, q_i^P), \quad (2.7)$$

where $p_u^P = f(p_u)$, $q_i^P = f(q_i + q_i^d)$.

with $f(\cdot)$ defined as:

$$f(x) = \phi(Wx + b), \quad (2.8)$$

where $\phi(\cdot)$ is an activation function. Since RnR is a general framework, any ranking/rating model can be used as f_{rank} and f_{rating} . For example, they implemented BPR (Rendle et al., 2009) and SVD (Paterek, 2007), leading to the following composite loss:

$$\mathcal{L} = \sum_{(u,i,j) \in \mathcal{R}} \left(-\alpha \cdot \ln \sigma(p_u^\top q_i - p_u^\top q_j) + (1 - \alpha) \cdot \left((p_u^P)^\top q_i^P - r_{ui} \right)^2 \right) \quad (2.9)$$

They compared RnR with a vanilla variant that employs shared embeddings across both implicit and explicit feedback:

$$\mathcal{L} = \sum_{(u,i,j) \in \mathcal{R}} \left(-\alpha \cdot \ln \sigma \left(\mathbf{p}_u^\top \mathbf{q}_i - \mathbf{p}_u^\top \mathbf{q}_j \right) + (1 - \alpha) \cdot \left(\mathbf{p}_u^\top \mathbf{q}_i - r_{ui} \right)^2 \right) \quad (2.10)$$

Hadash et al. (2018) reported only ranking results and therefore did not evaluate whether RnR also improved rating prediction. RnR achieved higher ranking performance than the vanilla variant that shared representations across the two objectives, providing evidence in their experimental setting for maintaining different pre- and post-consumption representations. Both RnR and the vanilla multi-task variant also outperformed the corresponding single-task models in ranking, indicating that the auxiliary rating objective contributed useful information to the ranking task. These experiments did not, however, establish whether the reverse relationship also held for rating prediction.

Since these earlier implicit-explicit models, research on learning from multiple forms of user behaviour has expanded considerably. Multi-behaviour recommender systems commonly preserve behaviour-specific information from actions such as views, clicks, add-to-cart events, and purchases while learning dependencies or transferring information across behaviour types (Kim et al., 2026). Recent graph-based approaches similarly construct behaviour-specific interaction structures and model cross-behaviour dependencies. This literature reinforces the broader motivation for maintaining heterogeneous feedback information rather than collapsing all signals into one representation. However, most multi-behaviour settings differ from the implicit-explicit setting considered in this thesis because they primarily use multiple implicit behaviours to improve prediction of a designated target behaviour, rather than jointly modelling behavioural interactions and numerical ratings for both ranking and rating prediction.

2.1.2 Single Feedback

Some methods have also assigned multiple representations to users or items, but only under a single type of feedback. Approaches that learn dual representations for items are often rooted in item-based CF, where a user’s rating for a target item is predicted based on their past ratings on similar items.

Several models adopt a common design in which each item has two embeddings: one is used when the item is the prediction target, while the other is used when the item contributes to the representation of a user who previously interacted with it. Under

this formulation, a user representation is constructed by aggregating embeddings of items in the user's interaction history. The models mainly differ in the aggregation operation, such as generalised pooling (Paterek, 2007; Kabbur et al., 2013) or attention (He et al., 2018b; Xue et al., 2019), and in the interaction function applied after the user and target-item representations are constructed (Kabbur et al., 2013; He et al., 2018b; Xue et al., 2019).

Paterek (2007) proposed **NSVD1**, a model designed for the rating prediction task using explicit feedback. In NSVD1, each item is assigned two embeddings, $\mathbf{p}$ and $\mathbf{q}$: $\mathbf{p}$ is used as the target item embedding, and $\mathbf{q}$ is utilised for constructing user representations. Formally, its predictive function is defined as:

$$\hat{y}_{ui} = \mathbf{p}_i^\top \left(\frac{1}{\sqrt{|\mathcal{R}_u^+| + 1}} \sum_{j \in \mathcal{R}_u^+} \mathbf{q}_j \right), \quad (2.11)$$

where $\mathcal{R}_u^+$ denotes the set of historical items rated by user u . In the same study, the authors proposed another model, NSVD2, which constructs the user representation using the same embedding set as the target items, instead of introducing an additional embedding set:

$$\hat{y}_{ui} = \mathbf{p}_i^\top \left(\sum_{j \in \mathcal{R}_u^+} \mathbf{p}_j \right). \quad (2.12)$$

Compared with NSVD1, NSVD2 uses the same item embedding set for both target-item prediction and user representation construction, reducing the number of item parameters from $O(2NK)$ to $O(NK)$. Paterek (2007) reported that NSVD1 achieved higher rating prediction accuracy than NSVD2 in their experiments, indicating that the additional role-specific item representation was beneficial in that setting.

Another influential model is the **FISM** (Factored Item Similarity Model) (Kabbur et al., 2013), which focuses on implicit feedback for the top-N recommendation task. In contrast to NSVD1, FISM explicitly excludes the target item i from the aggregation of historical items, $\mathcal{R}_u^+ \setminus \{i\}$, to avoid trivial self-similarity:

$$\hat{y}_{ui} = \mathbf{p}_i^\top \left(\frac{1}{|\mathcal{R}_u^+| - 1} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{q}_j \right), \quad (2.13)$$

where $\alpha \in [0, 1]$ is a hyperparameter interpolating between sum and average pooling.

Kabbur et al. (2013) reported that excluding the target item had relatively little effect at smaller embedding dimensions but produced clearer improvements at larger dimensions, supporting their decision to remove the target item from the user's aggregated representation.

Extending upon this approach, the **NAIS (Neural Attentive Item Similarity)** model (He et al., 2018b) also targets implicit feedback for top-N recommendation. Instead of generalised pooling, NAIS employs an attention mechanism to weight historical items differently according to their similarity to the target item:

$$\hat{y}_{ui} = \mathbf{p}_i^\top \left(\sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} a_{ij} \mathbf{q}_j \right), \quad (2.14)$$

where a_{ij} denotes the attention weight measuring item j 's contribution to user u 's representation for the target item i . Experiments demonstrated that NAIS achieved higher accuracy compared to FISM, validating the effectiveness of attention-based representations.

Building upon NAIS, the **DeepICF (Deep Item-based Collaborative Filtering)** model (Xue et al., 2019) adopts a similar user representation design but replaces the simple inner product interaction with a Multi-Layer Perceptron (MLP) to model more complex user-item interactions. Its formulation is:

$$\begin{aligned} \mathbf{e}_{ui} &= \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} a_{ij} (\mathbf{p}_i \odot \mathbf{q}_j), \\ \mathbf{e}_L &= \text{ReLU}(\mathbf{W}_L \text{ReLU}(\mathbf{W}_{L-1} \cdots \text{ReLU}(\mathbf{W}_1 \mathbf{e}_{ui} + \mathbf{b}_1) \cdots + \mathbf{b}_{L-1}) + \mathbf{b}_L), \end{aligned} \quad (2.15)$$

where the resulting vector $\mathbf{e}_{ui}$ is transformed through an L -layer MLP to obtain the final embedding $\mathbf{e}_L$. The interaction score $\hat{y}_{ui}$ is then computed by:

$$\hat{y}_{ui} = \mathbf{z}^\top \mathbf{e}_L, \quad (2.16)$$

with $\mathbf{z}$ being a learned parameter vector. Although DeepICF was shown to outperform FISM significantly, the authors did not explicitly compare its performance against NAIS, despite their methodological similarity.

2.2 Target-Aware and Interaction-Dependent Representation Learning

A fixed user or item embedding assumes that the same representation is equally suitable for every prediction target. Attention-based aggregation provides one way to relax this assumption by allowing different information to contribute to the representation with different weights. Of particular relevance to this thesis are target-aware approaches, in which representation construction depends on the current candidate item. The same user can therefore have different representations for different recommendation targets. This established line of work provides the closest conceptual background for the interaction-dependent representations investigated in Chapter 4. The term "dynamic" in DRL should be distinguished from temporal dynamics, where representations evolve over time.

The modern attention mechanism was initially developed in Neural Machine Translation to avoid compressing an entire input sequence into a single fixed context vector (Bahdanau et al., 2015). In its general query-key-value formulation, attention produces a weighted combination of value representations, with weights determined by the compatibility between a query and the corresponding keys (Vaswani et al., 2017). In recommender systems, related weighting mechanisms have been used to aggregate neighbours, behavioural histories, reviews, and other information when constructing user and item representations.

Not all attention-based recommendation representations are target-aware. For example, HDE (Luo et al., 2019) learned attention weights over neighbouring items when constructing a user representation and over neighbouring users when constructing an item representation. The attention scores were determined by the individual neighbour embeddings and learned parameters rather than by the current recommendation target:

$$u_i = \sum_{j \in \mathcal{R}_i} \alpha_j \cdot e_j, \quad \alpha_j = \frac{\exp(\alpha'_j)}{\sum_{y \in \mathcal{R}_i} \exp(\alpha'_y)}, \quad \alpha'_j = \alpha^\top \tanh(W_\alpha e_j + b_\alpha), \quad (2.17)$$

$$v_j = \sum_{i \in \mathcal{R}_j} \beta_i \cdot e_i, \quad \beta_i = \frac{\exp(\beta'_i)}{\sum_{x \in \mathcal{R}_j} \exp(\beta'_x)}, \quad \beta'_i = \beta^\top \tanh(W_\beta e_i + b_\beta), \quad (2.18)$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ denote the ID embeddings of user i and item j , respectively. Because the weighting function does not depend on the candidate counterpart, the resulting user or item representation remains unchanged across different target pairs once the model parameters are fixed.

Target-aware recommendation instead conditions representation construction on the current prediction target. In item-based formulations, a user's representation can be constructed as a weighted combination of items in the user's interaction history, where the contribution of each item depends on its relevance to the current candidate item. The same user can therefore have different representations for different candidates. NAIS (He et al., 2018b), DeepICF+a (Xue et al., 2019), and DIN (Zhou et al., 2018a) are representative examples of this principle.

One representative model is **NAIS** (He et al., 2018b) (see Equation 2.14 for its final prediction function), which assigns two embeddings $\mathbf{p}$ and $\mathbf{q}$ for each item. Given user u , the set of items in the user's interaction history $j \in \mathcal{R}_u^+$, and target item i , NAIS constructs the target-aware user embedding $\mathbf{e}_u(i)$ as:

$$\begin{aligned} \mathbf{e}_u(i) &= \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} a_{ij} \mathbf{q}_j, \\ a_{ij} &= \frac{\exp(f(\mathbf{p}_i, \mathbf{q}_j))}{\left[\sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \exp(f(\mathbf{p}_i, \mathbf{q}_j)) \right]^\beta}, \end{aligned} \quad (2.19)$$

where $f(\mathbf{p}_i, \mathbf{q}_j)$ measures the similarity between item i and item j , and $\beta \in [0, 1]$ is a smoothing exponent introduced to reduce the penalisation of attention weights for highly active users with long histories. NAIS offers two options for calculating similarity:

- Concatenation method:

$$f_{\text{concat}}(\mathbf{p}_i, \mathbf{q}_j) = \mathbf{h}^\top \text{ReLU}(\mathbf{W} [\mathbf{p}_i; \mathbf{q}_j] + \mathbf{b}), \quad (2.20)$$

- Element-wise product method:

$$f_{\text{prod}}(\mathbf{p}_i, \mathbf{q}_j) = \mathbf{h}^\top \text{ReLU}(\mathbf{W}(\mathbf{p}_i \odot \mathbf{q}_j) + \mathbf{b}), \quad (2.21)$$

where $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{h}$ are learned parameters. He et al. (2018b) reported similar final recommendation accuracy for the two attention formulations, while the element-wise

product formulation converged faster. NAIS also outperformed FISM (Kabbur et al., 2013) in their experiments, supporting the use of target-aware weighting rather than uniform aggregation of items in the user's interaction history.

Building on this design, **DeepICF+a** (Xue et al., 2019) used the same target-aware user representation principle as NAIS while applying a neural interaction function. The authors also evaluated DeepICF, a corresponding variant without the attention mechanism. Xue et al. (2019) reported that DeepICF+a outperformed DeepICF, providing additional evidence that target-aware weighting benefited their ranking model.

As a successfully deployed model in the industrial e-commerce setting, **DIN** (Deep Interest Network) (Zhou et al., 2018a) introduced a powerful mechanism for modelling user behaviour adaptively with respect to a candidate item. The model concatenates embeddings of various feature types, including user profile features (e.g., age level), context features (e.g., time), user behaviour features (i.e., items in the user's interaction history), and the candidate item, and feeds this combined input through an MLP to generate the final prediction score. For modelling user behaviour, DIN employs a mechanism similar to NAIS and DeepICF+a, using a weighted sum over the user's historical item embeddings, where the weights are adaptively computed based on the target item. Formally, given a list of embeddings of user u 's previously interacted items with length H , denoted by $\{e_1, e_2, \dots, e_H\}$, and a candidate item i , the dynamic user representation is calculated as:

$$e_u(i) = f(e_i, e_1, e_2, \dots, e_H) = \sum_{j=1}^H a(e_i, e_j) e_j, \quad (2.22)$$

where e_i and e_j represent item embeddings, which are themselves concatenations of multiple identifiers (e.g., goods ID, shop ID, and category ID). The attention score $a(e_i, e_j)$ is computed via a feed-forward network that estimates the relevance between the target item and each item in the user's interaction history:

$$\begin{aligned} a(e_i, e_j) &= h^\top \text{PreLU}(W [e_i; e_j; z] + b), \\ z &= \text{Flatten}(e_i e_j^\top), \end{aligned} \quad (2.23)$$

where z is the flattened outer product of e_i and e_j , capturing pairwise interactions between feature dimensions. Notably, unlike conventional attention mechanisms, DIN omits softmax normalisation in order to preserve the intensity and numerical resolution of user representations. Zhou et al. (2018a) reported that DIN outperformed the

compared baselines, including a variant that retained the broader model architecture but replaced the target-aware activation unit with sum pooling. This ablation provided evidence that conditioning behavioural aggregation on the candidate item contributed to DIN's performance in their industrial recommendation setting.

Guan et al. (2019) proposed an **AARM** (Attentive Aspect-based Recommendation Model), which constructs target-aware user representations at the aspect level. In their framework, aspects refer to words or phrases extracted from textual reviews that describe item features. Specifically, for user u and item v , the sets of aspects mentioned in their respective reviews are denoted by $\mathcal{A}_u$ and $\mathcal{A}_v$. Each aspect embedding is represented by a vector $\mathbf{c}$. For each aspect $i \in \mathcal{A}_u$, the model first calculates its interaction with all aspects in $\mathcal{A}_v$:

$$\mathbf{h}_{iv} = \sum_{j \in \mathcal{A}_v} \beta_{ij} (\mathbf{c}_i \odot \mathbf{c}_j), \quad (2.24)$$

where the aspect-level attention weight β_{ij} is defined as:

$$\begin{aligned} \hat{\beta}_{ij} &= \mathbf{w}_1^\top (\mathbf{c}_i \odot \mathbf{c}_j), \\ \beta_{ij} &= \frac{\exp(\hat{\beta}_{ij})}{\sum_{y \in \mathcal{A}_v} \exp(\hat{\beta}_{iy})}. \end{aligned} \quad (2.25)$$

Here $\mathbf{w}_1$ represents a learned parameter vector. Since different items may have different aspects, this formulation generates an aspect-level embedding $\mathbf{h}_{iv}$ for each user's aspect, dynamically varying with the target item v . These embeddings are then aggregated to construct the user-level dynamic embedding, where the item's aspects serve as the query:

$$\mathbf{y}_{uv} = \sum_{i \in \mathcal{A}_u} \alpha_{iv} \mathbf{h}_{iv} \quad (2.26)$$

with the user-level attention weight α_{iv} computed as:

$$\begin{aligned} \hat{\alpha}_{iv} &= \mathbf{w}_2^\top (\mathbf{c}_i \odot \sum_{j \in \mathcal{A}_v} \mathbf{c}_j), \\ \alpha_{iv} &= \frac{\exp(\hat{\alpha}_{iv})}{\sum_{x \in \mathcal{A}_u} \exp(\hat{\alpha}_{xv})}, \end{aligned} \quad (2.27)$$

where $\mathbf{w}_2$ is another learned parameter vector. To evaluate AARM's effectiveness, the authors compared it with two variants:

- A-Static, where the query for user-level attention uses all aspects from the user

themselves instead of the item (i.e., replacing $\sum_{j \in \mathcal{A}_v} c_j$) with $\sum_{j \in \mathcal{A}_u} c_j$) in Equation 2.27), thus constructing a static user embedding;

- No-UserAtt, which removes the user-level attention mechanism entirely, simplifying Equation 2.26 to:

$$y_{uv} = \sum_{i \in \mathcal{A}_u} h_{iv}. \quad (2.28)$$

Their experimental results showed that AARM significantly outperformed these variants, demonstrating once again the effectiveness of dynamically adjusting user representations according to the target item.

Target-aware representation learning has subsequently been explored beyond item-based CF and aspect-level recommendation. RTANet (Cao et al., 2023) constructed user embeddings whose neighbour aggregation weights depended explicitly on the recommendation target. MONET (Kim et al., 2024) introduced target-aware attention into multimedia recommendation, using the multi-modal features of the candidate item to identify information in previously interacted items that was particularly relevant to that target. More broadly, recent context-aware recommendation research continues to investigate how contextual information can be incorporated into user and item representations and prediction functions (Mateos and Bellogín, 2024). These developments show that target-aware representation is an established and continuing research direction rather than a concept introduced by this thesis.

Taken together, these studies show that representation construction can benefit from conditioning on the current recommendation target. The strongest evidence in the literature reviewed here concerns target-aware user representations, where different parts of a user’s behavioural history or aspect information are emphasised for different candidate items. Target-aware representation itself is therefore not the research gap addressed by this thesis.

The problem investigated in Chapter 4 is narrower. First, DRL instead considers interaction dependence on both sides of the user-item interaction, constructing an item-dependent user representation and a user-dependent item representation. Second, it conditions these representations not only on entity-level interaction histories but also on fine-grained aspect-sentiment information associated with review feedback. This setting introduces a circular dependency if the complete dynamic representation of

one entity is required to construct the representation of the other. DRL therefore separates general representations from interaction-specific information and uses the general representation of each counterpart to guide the construction of the other entity's interaction-dependent representation.

2.3 Aspect-Level Recommendation Models

Aspect-level recommendation requires textual reviews to be transformed into structured information about item characteristics and users' evaluations of those characteristics. Earlier aspect-level recommender systems commonly employed specialised sentiment-analysis pipelines such as Sentires⁵ to extract aspect, opinion, and sentiment information. More recent ABSA research encompasses transformer-based and generative language-model approaches and distinguishes explicitly expressed aspects from those inferred from context (Hua et al., 2024). This section focuses on how extracted aspect-level information is subsequently incorporated into recommendation models rather than on the extraction algorithm itself.

Let $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$, $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, and $\mathcal{A} = \{a_1, a_2, \dots, a_p\}$ denote the sets of m users, n items, and p aspects, respectively. In the formulation introduced by EFM (Zhang et al., 2014a), the extracted aspect and sentiment information is aggregated into a user-aspect attention matrix $\mathbf{X} \in \mathbb{R}^{m \times p}$ and an item-aspect quality matrix $\mathbf{Y} \in \mathbb{R}^{n \times p}$.

The element X_{ih} in $\mathbf{X}$ reflects the degree to which user u_i cares about aspect a_h , while the element Y_{jh} in $\mathbf{Y}$ measures the quality of item v_j with respect to aspect a_h . The matrices are computed as follows:

$$X_{ih} = \begin{cases} 0, & \text{if user } u_i \text{ does not mention aspect } a_h, \\ 1 + (N - 1)\left(\frac{2}{1 + e^{-t_{ih}}} - 1\right), & \text{otherwise,} \end{cases} \quad (2.29)$$

$$Y_{jh} = \begin{cases} 0, & \text{if aspect } a_h \text{ of item } v_j \text{ is not mentioned,} \\ 1 + \frac{N-1}{1 + e^{-s_{jh}}}, & \text{otherwise.} \end{cases} \quad (2.30)$$

Here, t_{ih} denotes the frequency with which user u_i mentions aspect a_h across all their reviews, and s_{jh} denotes the aggregated sentiment polarity towards aspect a_h across all reviews of item v_j . The values in both matrices are rescaled to fall within the range

⁵<https://github.com/evison/Sentires>

$[1, N]$, where N is the maximum allowable rating on the platform (e.g., $N = 5$ on Amazon and Yelp). In this way, both users and items are represented at the aspect level, facilitating fine-grained modelling of user preferences and item characteristics.

Based on $\mathbf{X}$ and $\mathbf{Y}$, as well as the overall user-item rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, Zhang et al. (2014a) proposed a hybrid matrix factorisation framework named **EFM** (Explicit Factor Model) for both rating prediction and top-N recommendation tasks. Specifically, the user-aspect attention matrix $\mathbf{X}$ and the item-aspect quality matrix $\mathbf{Y}$ share a common aspect-factor matrix $\mathbf{A} \in \mathbb{R}_+^{p \times k_1}$, which maps the p aspects into k_1 explicit latent dimensions. Accordingly, the user and item explicit factors are represented by $\mathbf{U}_1 \in \mathbb{R}_+^{m \times k_1}$ and $\mathbf{V}_1 \in \mathbb{R}_+^{n \times k_1}$.

Recognising that explicit factors may not fully explain a user's rating behaviour, the model also introduces additional latent factors $\mathbf{U}_2 \in \mathbb{R}_+^{m \times k_2}$ and $\mathbf{V}_2 \in \mathbb{R}_+^{n \times k_2}$ to capture hidden user preferences and item characteristics. Consequently, the user-factor matrix $\mathbf{U} \in \mathbb{R}_+^{m \times (k_1 + k_2)}$ and item-factor matrix $\mathbf{V} \in \mathbb{R}_+^{n \times (k_1 + k_2)}$ are constructed as:

$$\mathbf{U} = [\mathbf{U}_1 \ \mathbf{U}_2], \quad \mathbf{V} = [\mathbf{V}_1 \ \mathbf{V}_2]. \quad (2.31)$$

These components are jointly optimised through the following loss function:

$$\mathcal{L} = \|\hat{\mathbf{R}} - \mathbf{R}\|_F^2 + \lambda_x \|\hat{\mathbf{X}} - \mathbf{X}\|_F^2 + \lambda_y \|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2, \quad (2.32)$$

where the reconstructed matrices are defined as:

$$\hat{\mathbf{R}} = \mathbf{U}_1 \mathbf{V}_1^\top + \mathbf{U}_2 \mathbf{V}_2^\top, \quad \hat{\mathbf{X}} = \mathbf{U}_1 \mathbf{A}^\top, \quad \hat{\mathbf{Y}} = \mathbf{V}_1 \mathbf{A}^\top. \quad (2.33)$$

Here, λ_x and λ_y are hyperparameters that regulate the contribution of aspect-based reconstruction terms relative to the rating prediction loss.

The predicted rating for user u_i on item v_j is given by $\hat{R}_{ij}$. For top-N recommendation, the model assumes that users primarily care about a subset of aspects when deciding whether to engage with an item. Let $\mathcal{A}_i = \{a_{i1}, a_{i2}, \dots, a_{ik}\}$ denote the k most important aspects for user u_i , i.e., those corresponding to the top k values in the row vector $\hat{\mathbf{X}}_i$. Then, the ranking score of item v_j for user u_i is computed as:

$$\hat{s}_{ij} = \alpha \cdot \frac{\sum_{a \in \mathcal{A}_i} \hat{\mathbf{X}}_{i\langle a \rangle} \cdot \hat{\mathbf{Y}}_{j\langle a \rangle}}{kN} + (1 - \alpha) \cdot \hat{R}_{ij}, \quad (2.34)$$

where $\langle a \rangle$ denotes the position index of aspect a , $\alpha \in [0, 1]$ is a trade-off parameter that balances between the aspect-level matching score and the overall predicted rating.

Zhang et al. (2014a) reported that EFM outperformed the compared rating-prediction baselines that did not use review information. With an appropriate number of explicit factors k_1 , EFM also outperformed HFT (McAuley and Leskovec, 2013) in their experiments. For top-N recommendation, the authors observed that performance declined when the number k of user-preferred aspects became too large, indicating that including an increasingly broad set of aspects did not continue to improve ranking quality and could introduce less relevant aspect information.

Earlier aspect-level formulations such as EFM (Zhang et al., 2014a) modelled extracted aspects and their associated sentiment information but did not explicitly incorporate the corresponding opinion phrases into the learned recommendation representations. Distinguishing, for example, between *(screen, large, +1)* and *(screen, high-definition, +1)* can provide more detailed information about how users describe an aspect. To model this additional information, Wang et al. (2018) proposed a joint tensor factorisation framework, termed **MTER** (Multi-Task Explainable Recommendation), which integrates two complementary tasks: top-N recommendation and opinionated aspect-level explanation. The primary task is modelled by a three-dimensional tensor $\mathcal{X} \in \mathbb{R}_+^{m \times n \times p}$ over users, items, and aspects:

$$\mathcal{X}_{ijh} = \begin{cases} 0, & \text{if user } u_i \text{ does not mention aspect } a_h \text{ of item } v_j, \\ 1 + \frac{N-1}{1+e^{-s_{ijh}}}, & \text{otherwise,} \end{cases} \quad (2.35)$$

where s_{ijh} denotes the aggregated sentiment polarity expressed by u_i toward aspect a_h of item v_j across all reviews. To unify rating and aspect modelling, the authors introduced the overall rating as a dummy aspect, appending the rating matrix $\mathbf{R} \in \mathbb{R}_+^{m \times n}$ to form an extended tensor $\tilde{\mathcal{X}} \in \mathbb{R}_+^{m \times n \times (p+1)}$. The companion task models user and item opinion preferences via two additional tensors:

- $\mathcal{Y}^U \in \mathbb{R}_+^{m \times p \times q}$ over users, aspects, and opinion phrases, and
- $\mathcal{Y}^I \in \mathbb{R}_+^{n \times p \times q}$ over items, aspects, and opinion phrases.

These tensors are defined as:

$$\mathbf{y}_{ihw}^U = \begin{cases} 0, & \text{if } u_i \text{ does not express positive sentiment on } a_h \text{ via } o_w, \\ 1 + (N-1)\left(\frac{2}{1+e^{-t_{ihw}}} - 1\right), & \text{otherwise,} \end{cases} \quad (2.36)$$

$$\mathbf{y}_{jhw}^I = \begin{cases} 0, & \text{if } v_j \text{ receives no positive mention of } a_h \text{ via } o_w, \\ 1 + (N-1)\left(\frac{2}{1+e^{-t_{jhw}}} - 1\right), & \text{otherwise,} \end{cases} \quad (2.37)$$

where t_{ihw} and t_{jhw} denote the frequency of positive sentiment expressions toward aspect a_h using opinion phrase o_w by user u_i and toward item v_j , respectively. The overall objective incorporates a reconstruction loss on the three tensors, along with a BPR-style pairwise loss (Rendle et al., 2009) on the extended tensor for ranking:

$$\begin{aligned} \mathcal{L} = & \left\| \hat{\mathbf{x}} - \tilde{\mathbf{x}} \right\|_F^2 + \left\| \hat{\mathbf{y}}^U - \mathbf{y}^U \right\|_F^2 + \left\| \hat{\mathbf{y}}^I - \mathbf{y}^I \right\|_F^2 \\ & - \lambda \sum_{i=1}^m \sum_{(j,l) \in D_i} \ln \sigma \left(\hat{\mathbf{x}}_{ij(p+1)} - \hat{\mathbf{x}}_{il(p+1)} \right), \end{aligned} \quad (2.38)$$

where tensor reconstruction is computed via Tucker decomposition:

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{g}_1 \times_a \mathbf{U} \times_b \mathbf{I} \times_c \tilde{\mathbf{A}}, \\ \hat{\mathbf{y}}^U &= \mathbf{g}_2 \times_a \mathbf{U} \times_c \mathbf{A} \times_d \mathbf{O}, \\ \hat{\mathbf{y}}^I &= \mathbf{g}_3 \times_b \mathbf{I} \times_c \mathbf{A} \times_d \mathbf{O}. \end{aligned} \quad (2.39)$$

Here, $\times_a$ denotes the mode- a tensor-matrix product, and $\mathbf{g}_1 \in \mathbb{R}_+^{a \times b \times c}$, $\mathbf{g}_2 \in \mathbb{R}_+^{a \times c \times d}$, $\mathbf{g}_3 \in \mathbb{R}_+^{b \times c \times d}$ are core tensors. Matrices $\mathbf{U} \in \mathbb{R}_+^{m \times a}$, $\mathbf{I} \in \mathbb{R}_+^{n \times b}$, $\mathbf{A} \in \mathbb{R}_+^{p \times c}$, and $\mathbf{O} \in \mathbb{R}_+^{q \times d}$ denote latent factors for users, items, aspects, and opinions, respectively.

Wang et al. (2018) reported that opinion phrase usage varied across users and item aspects in their analysed data, motivating the tensor-based modelling of opinion information. In their ranking experiments, MTER also outperformed EFM. These results were consistent with the usefulness of modelling opinion phrases in addition to aspect-level information, although the two models differ in other architectural respects and the comparison therefore does not isolate opinion modelling as the sole cause of the improvement.

Luo et al. (2019) did not directly leverage aspect or sentiment information. Instead, they introduced temporal user-aspect attention matrices $\mathbf{X}^{(t)}$ and item-aspect quality

matrices $\mathbf{Y}^{(t)}$ as supervision signals. Reviews were encoded at the review level using a Doc2Vec approach (Lau and Baldwin, 2016), and temporal user preferences and item characteristics were captured through LSTM networks (Graves, 2013) over sequences of review vectors:

$$\mathbf{h}_i^{(t)} = \text{LSTM}_i \left(\mathbf{r}_i^{(t)} \right), \quad \mathbf{k}_j^{(t)} = \text{LSTM}_j \left(\mathbf{r}_j^{(t)} \right) \quad (2.40)$$

where $\mathbf{r}_i^{(t)}$ and $\mathbf{r}_j^{(t)}$ are the pretrained review vectors of user u_i and item v_j at timestep $t \in \{1, 2, \dots, T\}$, respectively.

The hidden states $\mathbf{h}_i^{(t)}$ and $\mathbf{k}_j^{(t)}$ were passed through fully connected layers to generate predictions $\hat{\mathbf{x}}_i^{(t)}$ and $\hat{\mathbf{y}}_j^{(t)}$, supervised by $\mathbf{X}^{(t)}$ and $\mathbf{Y}^{(t)}$:

$$\begin{aligned} \hat{\mathbf{x}}_i^{(t)} &= \sigma \left(\mathbf{W}_i \mathbf{h}_i^{(t)} + \mathbf{b}_i \right), \\ \hat{\mathbf{y}}_j^{(t)} &= \sigma \left(\mathbf{W}_j \mathbf{k}_j^{(t)} + \mathbf{b}_j \right) \end{aligned} \quad (2.41)$$

The overall objective combines rating prediction and auxiliary supervision:

$$\mathcal{L} = \sum_{(i,j) \in \mathcal{R}} \left\{ (\hat{r}_{ij} - r_{ij})^2 + \lambda \sum_{t=1}^T \left[\left(\hat{\mathbf{x}}_i^{(t)} - \mathbf{X}_i^{(t)} \right)^2 + \left(\hat{\mathbf{y}}_j^{(t)} - \mathbf{Y}_j^{(t)} \right)^2 \right] \right\}. \quad (2.42)$$

The supervision signals $\mathbf{X}^{(t)}$ and $\mathbf{Y}^{(t)}$ were constructed following Equation 2.29 by incorporating time-specific reviews. In addition, the model incorporated attention-based user and item embeddings, $\mathbf{u}_i$ and $\mathbf{v}_j$ (see Equations 2.17 and 2.18), and concatenated them with the final LSTM outputs to obtain:

$$\mathbf{u}'_i = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{h}_i^{(T)} \end{bmatrix}, \quad \mathbf{v}'_j = \begin{bmatrix} \mathbf{v}_j \\ \mathbf{k}_j^{(T)} \end{bmatrix}. \quad (2.43)$$

The final rating prediction is then computed via:

$$\hat{r}_{ij} = \text{ReLU} \left(\mathbf{W} \left(\mathbf{u}'_i \odot \mathbf{v}'_j \right) + \mathbf{b} \right) \quad (2.44)$$

Luo et al. (2019) reported that HDE outperformed EFM and the compared review- and topic-level baselines in rating prediction. HDE differed from these approaches by combining temporal review representations, aspect-level supervision, neighbourhood aggregation, and nonlinear interaction modelling. The reported comparisons therefore

showed that this combined architecture was effective in their experimental setting, but did not isolate aspect-level supervision as the sole source of the performance improvement.

2.4 Explanation Methods and Forms

Explainable recommender systems can provide explanations in different forms depending on the information represented by the underlying model. Aspect-based models may present structured item characteristics or fill sentence templates, counterfactual methods identify changes that would alter a recommendation, and generative models can produce complete natural-language explanations. The following discussion first illustrates aspect-level explanation mechanisms that are particularly relevant to this thesis and then briefly positions them relative to more recent counterfactual and language-model-based approaches.

The **EFM** model (Zhang et al., 2014a) is capable of identifying the top- k aspects $\mathcal{A}_i = \{a_{i1}, a_{i2}, \dots, a_{ik}\}$ that user u_i cares about the most. For a recommended item v_j , the aspect with the highest predicted quality score is selected as the basis for explanation:

$$a_{ij} = \operatorname{argmax}_{a \in \mathcal{A}_i} \hat{Y}_{j\langle a \rangle}. \quad (2.45)$$

Conversely, the model can also explain why an item is not recommended. For a disrecommended item, the aspect with the lowest quality score among the user's top aspects is selected:

$$a_{ij} = \operatorname{argmin}_{a \in \mathcal{A}_i} \hat{Y}_{j\langle a \rangle}. \quad (2.46)$$

Based on the selected aspect, the authors employed sentence templates to generate recommendation and disrecommendation explanations, respectively:

You might be interested in **[aspect]**, on which this product performs well.

You might be interested in **[aspect]**, on which this product performs poorly.

In addition to sentence-level explanations, the authors also presented visual explanations in the form of a word cloud, where positive aspect-opinion pairs are shown in green, negative ones in blue, and the font size is proportional to the term frequency in the item's reviews.

The **MTER** model (Wang et al., 2018) predicts an affinity tensor $\hat{\mathcal{X}} \in \mathbb{R}_+^{m \times n \times (p+1)}$, which encodes user-item-aspect interactions. Given a user u_i and a recommended item v_j , the model identifies the top- k_1 aspects with the highest affinity scores in $\hat{\mathcal{X}}_{ij}$, denoted as $\mathcal{A}_{ij} = \{a_{ij1}, a_{ij2}, \dots, a_{ijk_1}\}$. For each aspect $a \in \mathcal{A}_{ij}$, the model further predicts opinionated phrase distributions from both the user and the item sides: $\hat{\mathbf{y}}_{i\langle a \rangle}^U \in \mathbb{R}_+^q$ (user-side) and $\hat{\mathbf{y}}_{j\langle a \rangle}^I \in \mathbb{R}_+^q$ (item-side). The element-wise product of these two vectors yields an opinion phrase scoring vector:

$$\hat{\mathbf{y}}_{ij\langle a \rangle} = \hat{\mathbf{y}}_{i\langle a \rangle}^U \odot \hat{\mathbf{y}}_{j\langle a \rangle}^I \in \mathbb{R}_+^q \quad (2.47)$$

where the top- k_2 opinion phrases with the highest scores in $\hat{\mathbf{y}}_{ij\langle a \rangle}$ are selected to describe each aspect $a \in \mathcal{A}_{ij}$, denoted as $\mathcal{O}_{ij\langle a \rangle} = \{a_{ij\langle a \rangle 1}, a_{ij\langle a \rangle 2}, \dots, a_{ij\langle a \rangle k_2}\}$. Based on the selected aspects and associated opinion phrases, MTER generates opinionated aspect-level explanations. The following examples, generated from the Amazon (product: *Superleggera/Dual/Layer/Protection/Case*) and Yelp (restaurant: *Smash/Kitchen and Bar*) datasets with $k_1 = k_2 = 3$, illustrate the output:

Its **grip** is [firmer] [soft] [rubbery]. Its **quality** is [sound] [sturdy] [smooth]. Its **cost** is [original] [lower] [monthly].

Its **decor** is [neat] [good] [nice]. Its **sandwich** is [grilled] [cajun] [vegan]. Its **sauce** is [good] [green] [sweet].

A different explanation paradigm is **counterfactual explanation**. Rather than describing characteristics associated with the current recommendation, CountER (Tan et al., 2021) identifies a minimal set of item-aspect changes that would cause a recommended item to leave the recommendation list. The altered aspects therefore explain the recommendation in terms of features that are sufficient to change the model’s decision. Counterfactual approaches are particularly useful for evaluating explanation faithfulness because their effects can be tested directly by intervening on model inputs.

Natural-language generation has also become an increasingly prominent explanation form. PEPLER (Li et al., 2023) used personalised prompt learning with a pre-trained language model to generate recommendation explanations. More recent work has incorporated large language models. LLM2ER (Yang et al., 2024), for example, fine-tuned an LLM with explanation-quality reward signals, while XRec (Ma et al., 2024) aligned LLM generation with collaborative recommendation information. A recent

review found that direct research on LLM-generated recommendation explanations was still relatively limited, indicating that this remains an emerging research direction (Said, 2025). These methods can produce fluent user-facing explanations, but linguistic fluency alone does not establish that a generated statement faithfully reflects the evidence used by the underlying recommender.

2.5 Evaluation of Explanations

Explanation evaluation methods can be broadly grouped into online evaluations, user studies, offline quantitative evaluations, and qualitative case studies. These approaches assess different properties of an explanation and should not be regarded as interchangeable. Online evaluations measure effects on observable behaviour in deployed systems, user studies assess human perceptions or decision outcomes under controlled conditions, offline metrics evaluate measurable proxies such as agreement with held-out evidence or model faithfulness, and case studies provide qualitative inspection of representative examples. In particular, strong performance on an offline explanation metric does not by itself establish that users will perceive the explanation as useful, understandable, persuasive, or trustworthy (Wardatzky et al., 2025).

2.5.1 Online Evaluations

Online evaluations assess explanation effects through user behaviour in deployed systems. Zhang et al. (2014a) conducted two A/B tests on the online shopping website JingDong⁶. In the first A/B test, users were assigned to three conditions: an experimental group receiving aspect-level explanations, a comparison group receiving "People also viewed" recommendations, and a control group receiving no explanations. The authors reported a higher Click-Through Rate (CTR) for the aspect-explanation condition than for the two comparison conditions. In the second A/B test, the authors evaluated add-to-cart behaviour when recommendation and disrecommendation explanations were displayed. They reported that the explanation condition affected conversion behaviour and that disrecommendation explanations reduced add-to-cart actions for unsuitable items.

⁶<https://www.jd.com>

2.5.2 User Studies

User studies collect human judgements about explanations under controlled conditions. Wang et al. (2018) presented participants with a user's previous reviews, a recommendation, and its explanation, and asked whether the explanation helped them learn about the recommended item and understand the recommender system. The participants gave more favourable responses to opinionated aspect-level explanations than to aspect-only explanations. Xiong et al. (2021) instead used a pairwise preference design, asking participants to compare their proposed counterfactual explanations with aspect-level explanations generated by A2CF (Chen et al., 2020b) in terms of perceived helpfulness.

2.5.3 Offline Formula-Based Methods

Offline formula-based methods evaluate explanations using predefined quantitative criteria without directly collecting user judgements. Some methods compare predicted explanatory features with evidence contained in held-out reviews, whereas others use proxy criteria derived from the recommendation model itself. When held-out reviews are used as reference data, metrics such as Precision, Recall, or NDCG measure whether predicted aspects or opinion phrases also appear in the user's observed review. These measures quantify agreement with the selected reference evidence but do not directly measure the user's perception of explanation quality.

Wang et al. (2018) used **NDCG** (Normalised Discounted Cumulative Gain) metric to verify if the predicted top- k_1 aspects $\mathcal{A} = \{a_1, a_2, \dots, a_{k_1}\}$ and top- k_2 opinion phrases $\mathcal{O}_{\langle a \rangle} = \{a_{\langle a \rangle 1}, a_{\langle a \rangle 2}, \dots, a_{\langle a \rangle k_2}\}$ for each aspect $a \in \mathcal{A}$ are indeed mentioned in the user's review:

$$\text{NDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k}. \quad (2.48)$$

Here

$$\text{DCG}@k = \sum_{i=1}^k \frac{\text{rel}(i)}{\log_2(i+1)}, \quad (2.49)$$

where $\text{rel}(i)$ is the relevance score of the element at position i in the ranking list. The $\text{IDCG}@k$ is defined as "Ideal $\text{DCG}@k$ " and calculated by sorting the relevance scores in descending order under the ideal scenario (ground truth) and then computing $\text{DCG}@k$. The authors computed two NDCG scores: one for the aspect $\text{NDCG}@20$ and the other for the opinion phrase $\text{NDCG}@50$.

In their offline explanation evaluation, Wang et al. (2018) reported higher aspect-prediction NDCG than EFM (Zhang et al., 2014a). Because EFM did not predict opinion phrases, MTER's opinion predictions were compared with a popularity-based Random baseline, which MTER also outperformed.

Chen et al. (2020b) introduced a metric called **ATC** (Attribute Trade-off Coverage) to evaluate the quality of explanations in substitute recommendation scenarios. ATC is defined as the harmonic mean of Average Precision (AP) and NDCG:

$$\text{ATC} = \text{mean}_{\forall (u_i, v_q, v_j)} \frac{2 \cdot \text{AP} \cdot \text{NDCG}}{\text{AP} + \text{NDCG}}, \quad (2.50)$$

where (u_i, v_q, v_j) denotes a test instance in which user u_i eventually chooses item v_j after browsing item v_q . For each instance, the model generates an aspect ranking list $\mathcal{A}_r = \{a_1, a_2, \dots, a_r\}$, where r is the total number of aspects in the dataset. The AP score quantifies how well the predicted aspects match the ground-truth aspects $\mathcal{G}$ mentioned in user u_i 's review on item v_j :

$$\text{AP} = \frac{\sum_{k=1}^r \text{Precision@}k}{|\mathcal{G}|}, \quad (2.51)$$

where $|\mathcal{G}|$ is the number of ground-truth aspects, and

$$\text{Precision@}k = \frac{\sum_{t=1}^k \mathbf{I}(f_t \in \mathcal{G})}{k}, \quad (2.52)$$

with $\mathbf{I}(\cdot)$ being an indicator function that returns 1 if the condition holds and 0 otherwise.

Luo et al. (2019) introduced a metric called **GOFE** (Goodness Of Fit on Explanation) to evaluate the explainability of their model HDE. The method first identifies the top- p aspects learned for each user u , denoted as $\mathcal{A}_p^{(u)} = \{a_1^{(u)}, a_2^{(u)}, \dots, a_p^{(u)}\}$, and then determines the top- q items associated with each aspect $a \in \mathcal{A}_p^{(u)}$, denoted as $\mathcal{V}_q^{(a)} = \{v_1^{(a)}, v_2^{(a)}, \dots, v_q^{(a)}\}$. An indicator function is defined as:

$$\mathbf{I}(u, v) = \begin{cases} 1 & \text{if } \exists a \in \mathcal{A}_p^{(u)}, v \in \mathcal{V}_q^{(a)} \\ 0 & \text{otherwise,} \end{cases} \quad (2.53)$$

where v denotes an item recommended to user u . Based on this, the GOF metric is computed as:

$$\text{GOF}@k, p, q = \frac{\sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{R}_k^{(u)}} \mathbf{I}(u, v)}{|\mathcal{U}| \times k}, \quad (2.54)$$

where $\mathcal{R}_k^{(u)} = v_1, v_2, \dots, v_k$ is the top- k recommended items for user u , and $\mathcal{U}$ is the set of target users. This metric can be interpreted as the likelihood that a recommended item is aligned with a user's most preferred aspects and is thus explainable. Because GOF does not compare explanations with held-out ground-truth aspects, it measures consistency between the modelled user preferences, item qualities, and recommendation list rather than agreement with an independently observed explanation. The metric is also sensitive to the manually selected values of p and q .

Tan et al. (2021) learnt a slight change vector $\Delta_{i,j} = (\delta_{i,j}^{(1)}, \delta_{i,j}^{(2)}, \dots, \delta_{i,j}^{(r)})$ whose values are either zero or negative and applied it to the original item-aspect quality vector $\mathbf{y}_j = (y_j^{(1)}, y_j^{(2)}, \dots, y_j^{(r)})$, i.e., $\mathbf{y}_j + \Delta_{i,j}$, to push the counterfactual item out of the top- N recommendation list $\mathcal{R}_{i,k}$ for user u_i , where r is the total number of aspects. Their aspect-level and counterfactual explanations consist of the aspects that have non-zero values in $\Delta_{i,j}$, i.e., $\mathcal{F}_{i,j}$. They used multiple metrics to assess the explanations from two perspectives: user-oriented and model-oriented. The user-oriented evaluation addresses whether the explanations align with the user's preferences. They treated all the aspects with positive sentiments in the user's review on the item as ground truth: $\mathbf{p}_{i,j} = (p_{i,j}^{(1)}, p_{i,j}^{(2)}, \dots, p_{i,j}^{(r)})$ is defined as a binary vector, where $p_{i,j}^{(t)}$ is 1 if user u_i has positive sentiment on the aspect a_t in their review for item v_j , 0 otherwise. They then selected three common top- k recommendation metrics to compare the learnt aspects to the ground truth:

$$\text{Precision} = \frac{\sum_{t=1}^r p_{i,j}^{(t)} \cdot \mathbf{I}(\delta_{i,j}^{(t)} \neq 0)}{\sum_{t=1}^r \mathbf{I}(\delta_{i,j}^{(t)} \neq 0)}, \quad (2.55)$$

$$\text{Recall} = \frac{\sum_{t=1}^r p_{i,j}^{(t)} \cdot \mathbf{I}(\delta_{i,j}^{(t)} \neq 0)}{\sum_{t=1}^r p_{i,j}^{(t)}}, \quad (2.56)$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (2.57)$$

where $\mathbf{I}(x)$ is an indicator function such that $\mathbf{I}(x) = 1$ when the condition x holds

and 0 otherwise. To test whether the explanations reveal the essential recommendation mechanism of the recommendation model, they proposed three model-oriented metrics: **PS** (Probability of Sufficiency), **PN** (Probability of Necessity) and F_{NS} . The idea of PS is to calculate the probability that item v_j would still be recommended to user u_i if $\mathcal{A}_{i,j}$ were the only aspects present in the system. Specifically, they changed the aspect values of all items in the item-aspect quality matrix Y to 0 if the aspects are not in $\mathcal{A}_{i,j}$ and checked whether the counterfactual item v'_j is still in the counterfactual top-k recommendation list $\mathcal{R}'_{i,k}$:

$$\text{PS} = \frac{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{R}_{i,k}} \mathbf{I}(v'_j \in \mathcal{R}'_{i,k})}{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{R}_{i,k}} \mathbf{I}(\mathcal{A}_{i,j} \neq \emptyset)}. \quad (2.58)$$

The idea of PN is to calculate the probability that item v_j would not be recommended to user u_i if $\mathcal{A}_{i,j}$ did not exist in the system. Similarly, they applied interventions by altering the aspect values of all items in the item-aspect quality matrix Y to 0 if the aspects are in $\mathcal{A}_{i,j}$ and checked whether the counterfactual item v^*_j would be removed from the counterfactual top-k recommendation list $\mathcal{R}^*_{i,k}$:

$$\text{PN} = \frac{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{R}_{i,k}} \mathbf{I}(v^*_j \notin \mathcal{R}^*_{i,k})}{\sum_{u_i \in \mathcal{U}} \sum_{v_j \in \mathcal{R}_{i,k}} \mathbf{I}(\mathcal{A}_{i,j} \neq \emptyset)}. \quad (2.59)$$

The PS and PN evaluate the extent that the generated aspects are sufficient and necessary, respectively. The F_{NS} is their harmonic mean value:

$$F_{NS} = 2 \cdot \frac{\text{PN} \cdot \text{PS}}{\text{PN} + \text{PS}}. \quad (2.60)$$

2.5.4 Case Studies

Case studies provide qualitative evidence by examining representative recommendation and explanation examples in detail. Luo et al. (2019) visualised user and item profiles as aspect word clouds and compared the alignment between one user's preferences and the qualities of recommended and disrecommended items. Chen et al. (2020b) presented three examples comparing predicted aspect-level explanations with aspects observed in users' reviews. Xiong et al. (2021) illustrated two examples containing users' positively and negatively rated items together with the features identified by their counterfactual model as requiring the smallest changes to reverse the

recommendation outcome.

Taken together, the literature reviewed in this chapter highlights two recurring representation-learning problems that motivate the remainder of this thesis. First, heterogeneous forms of feedback can provide complementary information, but existing approaches differ in whether feedback-specific representations should be shared, separated, or connected. Second, target-aware recommendation research demonstrates that one fixed user representation need not be appropriate for every interaction, while review-based and aspect-level approaches provide fine-grained information that can support both preference modelling and explanation. Chapters 3 and 4 investigate these two problems through feedback-dependent and interaction-dependent representation learning, respectively. The literature on explanation evaluation further shows that predictive agreement with explanatory evidence and user-perceived explanation quality are related but distinct evaluation objectives.

Chapter 3

IEL: Interdependent Embedding Learning

Personalised recommender systems aim to model user preferences and item characteristics from users' feedback. Feedback is commonly divided into implicit and explicit forms (Xie et al., 2020). **Implicit feedback**, such as clicks and purchases, records users' behavioural interactions with items. Such signals can usually be collected automatically at scale, but an observed interaction does not directly indicate the strength of a user's preference, and an unobserved interaction does not necessarily indicate dislike (Hu et al., 2008). Consequently, learning from implicit feedback requires dealing with ambiguous positive signals and the absence of explicit negative feedback. **Explicit feedback**, such as numerical ratings, allows users to express the degree of their preference more directly. However, collecting explicit feedback requires additional user effort and is therefore often sparser than behavioural interaction data (Jawaheer et al., 2010; Liu et al., 2010). Implicit and explicit feedback consequently provide complementary information: the former captures behavioural engagement, whereas the latter provides more direct evaluative information about the observed interaction.

The different characteristics of the two feedback types have led to different recommendation objectives (Wan and McAuley, 2018). **Ranking models** commonly learn from implicit interactions to produce a personalised ordering of candidate items, often referred to as top- N recommendation. Their objective is to rank items with which a user is more likely to interact above less relevant items, rather than to estimate an exact preference value (Cremonesi et al., 2010). Ranking performance is therefore commonly evaluated using information retrieval metrics such as Recall and NDCG (Bellogín et al., 2017). **Rating prediction models**, in contrast, learn primarily from explicit feedback and estimate the numerical rating that a user would assign to an item (Hadaş et al., 2018). This task is usually formulated as regression and evaluated using error-based

metrics such as RMSE or MSE (Steck, 2010). Ranking and rating prediction therefore capture related but different aspects of preference: ranking concerns the relative ordering of items, whereas rating prediction estimates the strength of a user’s explicit evaluation.

Although ranking and rating prediction have different objectives, previous research has shown that jointly modelling implicit and explicit feedback can improve recommendation performance (Liu et al., 2010; Jadidinejad et al., 2019). Existing unified approaches differ in how they represent the two feedback types and how the corresponding learning objectives are connected (Hadaş et al., 2018). Some methods (Liu et al., 2010; Shi et al., 2013; Jadidinejad et al., 2019) used the same user or item representations, and in some cases the same predicted score, for both implicit and explicit feedback. Such a formulation does not explicitly represent the possibility that the two feedback types convey different information about the same user-item relationship. For example, purchasing an item indicates behavioural engagement, whereas subsequently assigning it a low rating indicates an unfavourable explicit evaluation. These observations are not contradictory: they describe different stages or aspects of the user’s interaction with the item. This motivated us to represent users and items separately under implicit and explicit feedback while allowing information to be exchanged between the resulting representation spaces.

Some earlier models already assigned different representations to implicit and explicit feedback, either for items (Bell and Koren, 2007; Koren, 2008a; Liu et al., 2017; Chen and Peng, 2018) or for both users and items (Hadaş et al., 2018; Mandal and Maiti, 2020). Several of these methods also transferred neighbour information across feedback types. For example, SVD++ and related approaches incorporated the implicit embeddings of items previously interacted with by a user into that user’s explicit representation (Bell and Koren, 2007; Koren, 2008a; Liu et al., 2017). This represents a one-directional relationship from implicit item representations to an explicit user representation. A complementary relationship had received much less attention: an item’s implicit representation could also benefit from the explicit representations of users who evaluated that item. More generally, earlier methods did not investigate how this cross-feedback relationship between implicit item and explicit user representations could be combined with collaborative neighbourhood propagation within both feedback-specific representation spaces. This distinction motivated our method to treat feedback-specific representations as distinct but interdependent rather than

either completely shared or independently learned. We therefore investigated the following research question: **How can implicit and explicit feedback be integrated to learn user and item representations that preserve feedback-specific information and support both item ranking and rating prediction?**

Neighbour-based representation learning is naturally related to Graph Convolutional Networks (GCNs). In recommender systems, user-item interactions can be represented as bipartite graphs, and graph message passing allows users and items to aggregate information from multi-hop collaborative neighbourhoods (Zhang et al., 2024). GCN-based methods have been applied to both implicit-feedback ranking (Wang et al., 2019; He et al., 2020) and explicit-feedback recommendation (Zhang et al., 2019a; Zhang et al., 2023b). More recent work has also shown that explicit rating graphs can benefit from propagation mechanisms that preserve rating-specific information rather than treating all graph edges identically (Peng et al., 2025).

Related work has considered heterogeneous user actions from both micro-behaviour and multi-behaviour perspectives. Micro-behaviour research focuses on fine-grained operations within user-item interactions, such as how an item is accessed and whether a user reads comments or adds the item to a cart (Zhou et al., 2018b; Meng et al., 2020; Yuan et al., 2022). Multi-behaviour recommendation more broadly models multiple types of user-item behaviour, such as views, clicks, add-to-cart actions, and purchases, while preserving their different semantics and learning dependencies across behaviour types (Zhao et al., 2024a; Kim et al., 2026). These two perspectives overlap, since fine-grained user actions may themselves be treated as distinct behaviour types.

Our method was related to this broader principle of preserving heterogeneous interaction semantics while allowing information transfer between them, but it addressed a different recommendation setting. Multi-behaviour recommendation commonly uses auxiliary implicit behaviours such as clicks or add-to-cart actions to improve prediction of a designated target behaviour (Kim et al., 2026). In contrast, our method considered implicit and explicit feedback as two representation spaces associated with different recommendation objectives and jointly supported both item ranking and rating prediction. Earlier neighbour-augmentation approaches (Bell and Koren, 2007; Koren, 2008a; Liu et al., 2017; Pan et al., 2016) can be interpreted as first-order forms of such information transfer. Our method extended this idea by using graph propagation to model both within-feedback collaborative relationships and selected cross-feedback relationships over multiple propagation layers.

To investigate the research question, we developed a **GCN-based unified framework**, termed **Interdependent Embedding Learning (IEL)**, for jointly modelling implicit and explicit feedback. IEL maintained separate user and item representations for the two feedback types and constructed corresponding implicit and explicit user-item graphs. Intra-graph message passing captured collaborative relationships within each feedback type, while selected inter-graph pathways allowed information to pass between the feedback-specific representation spaces. In particular, the implicit item and explicit user representations formed the principal bridge between the two graphs, after which the transferred information could propagate further through their respective collaborative neighbourhoods. This architecture operationalised the hypothesis that feedback-specific representations should remain distinct while still being able to inform one another. IEL was evaluated on the Amazon Office and Video Games rating datasets. In these experiments, observed ratings provided the explicit feedback, while their corresponding observed user-item interactions were binarised to construct the implicit-feedback view. The experiments evaluated both item ranking and rating prediction. The main contributions of this study were:

1. We developed a unified recommendation framework that maintained separate user and item representations for implicit and explicit feedback while learning the ranking and rating prediction objectives jointly. The experimental results showed that modelling these feedback-specific representations as interdependent improved recommendation performance relative to the compared baselines.
2. Conceptually, we investigated a cross-feedback relationship that had received limited attention in earlier neighbour-based unified recommendation: the relationship between an implicit item representation and the explicit representations of users who evaluated that item. Together with the previously studied transfer from implicit items to explicit users, this allowed the two feedback-specific representation spaces to be treated as distinct but interdependent.
3. Technically, we implemented this interdependence through GCN-based intra-graph and inter-graph message passing. Cross-feedback information was first transferred through selected bridge representations and then propagated through collaborative neighbourhoods within the corresponding feedback graphs, allowing higher-order neighbour information to contribute to both ranking and rating prediction.

3.1 Related Work

Chapter 2 provided a broader review of collaborative filtering, graph-based recommendation, and approaches for learning from different forms of user feedback. The purpose of this section is more specific. We focus on models that jointly used implicit and explicit feedback and examine three questions that are directly relevant to IEL: whether feedback-specific user and item representations were shared or separated, how information was transferred between those representations, and whether the resulting model was evaluated for ranking, rating prediction, or both.

For this purpose, the related methods are organised according to their reported recommendation objectives: rating prediction, ranking recommendation, and unified evaluation. Table 3.1 summarises representative approaches and highlights whether user and item representations were shared across feedback types and how the two representation spaces were connected.

3.1.1 Rating Prediction

Early rating-prediction approaches already demonstrated that implicit interactions could complement explicit user representations. In SVD++ (Koren, 2008a), for example, a user's explicit representation incorporated the aggregated representations of items with which the user had implicitly interacted. Related neighbourhood-based approaches similarly used behavioural interactions as supplementary information for predicting explicit ratings (Bell and Koren, 2007). These methods established an important connection between the two feedback types, but the information transfer was primarily directed from implicit item representations towards the explicit user representation.

3.1.2 Ranking Recommendation

Shi et al. (2013) employed shared embeddings of users and items under implicit and explicit feedback. In their approach, the same embedding produced the same predicted score, which was used in both the ranking algorithm and the rating prediction algorithm. Although users/items had different embeddings in Liu et al. (2022)'s model, their implicit and explicit embeddings were concatenated immediately at the embedding layer to form a unified embedding. In both cases, the two feedback types were combined before producing the final ranking signal. Such designs enabled information

TABLE 3.1: Models utilising implicit and explicit feedback.

Model	$\mathbf{u}^{(im)} = \mathbf{u}^{(ex)}$	$\mathbf{v}^{(im)} = \mathbf{v}^{(ex)}$	Connection	Rec Obj
SVD++ (Koren, 2008a)	-	No	$\mathbf{u}^{(ex)} + = \text{AGG}(\mathbf{v}^{(im)})$	Rating
Co-Rating (Liu et al., 2010)	Yes	Yes	None	Both
URM (Shi et al., 2013)	Yes	Yes	None	Ranking
MFPR (Liu et al., 2017)	-	No	$\mathbf{u}^{(ex)} + = \frac{1}{\tau} \sum_{t=1}^{\tau} \text{AGG}(\mathbf{v}_t^{(im)})$	Ranking
PMF+P+N (Chen and Peng, 2018)	Yes	No	None	Both
RnR (Hadaş et al., 2018)	No	No	$\mathbf{u}^{(ex)} = \text{FC}(\mathbf{u}^{(im)})$ $\mathbf{v}^{(ex)} = \text{FC}(\mathbf{v}^{(im)} + \mathbf{v}^{(pos)})$	Ranking
Weak Supervision (Jadidinejad et al., 2019)	Yes	Yes	None	Both
GPMF (Mandal and Maiti, 2020)	No	No	$\min \ \mathbf{u}^{(ex)} - \mathbf{u}^{(im)}\ _F^2$	Both
ITE (Tran et al., 2022)	Yes	Yes	$\hat{s} = f(\Phi^{(im)})$ $\hat{r} = g(\text{MLP}(\Phi^{(im)}))$	Ranking
EINMF (Liu et al., 2022)	No	No	None	Ranking
IEL (our method)	No	No	$\mathbf{u}^{(im)} = \text{AGG}(\mathbf{v}^{(im)})$ $\mathbf{v}^{(im)} = \text{AGG}(\mathbf{u}^{(im)}, \mathbf{u}^{(ex)})$ $\mathbf{u}^{(ex)} = \text{AGG}(\mathbf{v}^{(ex)}, \mathbf{v}^{(im)})$ $\mathbf{v}^{(ex)} = \text{AGG}(\mathbf{u}^{(ex)})$	Both

sharing but limited the model's ability to preserve different meanings for behavioural engagement and explicit evaluation at the prediction stage.

Tran et al. (2022) also did not separate the user/item embeddings for different types of feedback. Instead, they considered the implicit and explicit feedback provided by users as a sequential process, with explicit scores derived based on implicit feedback information. Specifically, the vector that generated the implicit score passed through several Multi-Layer Perceptron (MLP) layers to obtain the explicit score. Although the same user/item embeddings were used in different feedback, this process simulated the relationship and difference between implicit and explicit feedback.

In Tran et al. (2022), the two feedback types were separated by first merging user and item embeddings to obtain an implicit vector, which was then transformed to derive the explicit vector. However, in Hadash et al. (2018), this transformation occurred on the user and item embeddings before they were merged. More specifically, the implicit user embedding and implicit item embedding were utilised to predict implicit feedback. Subsequently, a deviation item embedding was added to the implicit item embedding to derive the post item embedding. Finally, the implicit user embedding and the post item embedding were passed through a fully connected layer (FC) with a non-linear activation function to generate the explicit user embedding and explicit item embedding, respectively. The explicit embeddings generated in this manner were then merged for rating prediction.

Based on the design of SVD++ (Koren, 2008a), which utilised a single type of implicit item embedding to complement explicit user embedding, Liu et al. (2017) introduced multiple types of implicit item embeddings as supplementary information. However, the supplemented explicit user embedding and the explicit item embedding were integrated for item ranking rather than rating prediction. Their results showed that incorporating multiple forms of implicit interaction could improve ranking performance, further supporting the usefulness of transferring complementary feedback information across representation components.

3.1.3 Unified Tasks

An early approach that jointly learned from implicit and explicit feedback for both ranking and rating prediction was proposed by Liu et al. (2010). Like Shi et al. (2013), they utilised the same user and item embeddings and generated the same predicted score for both learning tasks. Their difference only lies in the choice of the ranking

algorithm. Although they utilised the same embeddings for both learning tasks, their work has shown that combining implicit and explicit feedback could improve recommendation quality.

Considering implicit feedback is more abundant than explicit feedback, Jadidinejad et al. (2019) used a rating prediction model trained on explicit feedback to label implicit feedback. Specifically, the original values of 1 in the implicit feedback were converted to predicted ratings from the rating prediction algorithm, thus creating a new weakly annotated dataset. The model was then trained on this dataset. Their approach utilised the same user/item embeddings across different feedback types, and the final predicted score was used for both ranking and rating prediction, similar to (Liu et al., 2010; Shi et al., 2013). The difference, however, was that they only employed the rating prediction algorithm for training, rather than a weighted sum of two tasks.

Chen and Peng (2018) assigned different embeddings to each item based on different feedback. They proposed that implicit and explicit feedback could correspond to the item display embedding and its true embedding, respectively. The shared user embedding is then utilised with the item display embedding for ranking and the item true embedding for rating prediction.

Mandal and Maiti (2020)'s model learnt different user/item embeddings from different types of feedback. In their approach, the user embeddings under different feedback types were constrained by an L2 norm in the loss function. The purpose was to ensure that while the user embeddings may differ across feedback types, they should be as close as possible.

3.1.4 Summary and Research Gap

The reviewed literature showed that implicit and explicit feedback had been connected in several different ways. One group of methods shared the same user/item representations or prediction scores across the two feedback types (Liu et al., 2010; Shi et al., 2013; Jadidinejad et al., 2019). This approach enabled joint learning but did not explicitly preserve the different semantics of behavioural engagement and explicit evaluation. A second group maintained different representations and connected them through transformations or regularisation (Hadash et al., 2018; Mandal and Maiti, 2020). These methods preserved some feedback-specific information, but the relationships between the representation spaces were primarily encoded through parameter transformations or loss constraints. A third group transferred neighbour information

between feedback types (Bell and Koren, 2007; Koren, 2008a; Liu et al., 2017; Pan et al., 2016), demonstrating that collaborative neighbours could provide complementary information across feedback types.

Taken together, these studies suggested two useful principles for unified recommendation. First, implicit and explicit feedback should not necessarily be forced into identical representations because they convey different information about user-item relationships. Second, keeping the representations completely independent would discard potentially useful relationships between the two forms of feedback. The remaining question was therefore how feedback-specific representations could remain distinct while exchanging collaborative information in a structured manner.

Recent multi-behaviour recommendation research has strengthened this general motivation by preserving behaviour-specific interaction structures and explicitly learning cross-behaviour dependencies (Zhao et al., 2024a; Kim et al., 2026). Nevertheless, most multi-behaviour settings use several implicit behaviours to improve prediction of a designated target behaviour, whereas the problem investigated in this chapter jointly considered implicit behavioural interactions, explicit ratings, and two corresponding objectives: ranking and rating prediction. IEL addressed this gap by using separate feedback-specific graphs and GCN-based message passing to connect the implicit item and explicit user representation spaces while allowing transferred information to propagate through their collaborative neighbourhoods.

3.2 Preliminaries and Model Motivation

Our GCN-based unified model integrates both implicit and explicit feedback to address two key recommendation tasks: item ranking and rating prediction. As such, it incorporates components from rating prediction algorithms, ranking algorithms, and GCN-based representation learning. In this section, we briefly review a classical rating prediction approach (Matrix Factorisation (Koren et al., 2009)), and a widely adopted ranking method (Bayesian Personalised Ranking (Rendle et al., 2009)), as well as the GCN framework. We further elaborate on the architectural motivation behind our model, highlighting the challenges involved in performing message passing both within (intra-graph) and across (inter-graph) feedback-specific interaction graphs.

3.2.1 Matrix Factorisation

Explicit feedback is typically in the form of numerical ratings provided by users to directly express their preferences. For instance, in a 1-5 rating system, a high rating indicates that the user likes the item, whereas a low rating suggests the opposite.

Let $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$ denote a user and an item, where m and n represent the number of users and items, respectively. The user-item rating matrix is defined as $\mathbf{R} \in \mathbb{R}^{m \times n}$. The goal of recommendation with explicit feedback is to predict a rating $\hat{r}_{i,j}$ that accurately reflects user i 's preference for item j .

Matrix Factorisation techniques are simple yet effective approaches for rating prediction. They decompose the original rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ into a user embedding matrix $\mathbf{U} \in \mathbb{R}^{d \times m}$ and an item embedding matrix $\mathbf{V} \in \mathbb{R}^{d \times n}$, where d is the dimension of the user/item embeddings. The user embedding $\mathbf{u}_i \in \mathbb{R}^d$ and item embedding $\mathbf{v}_j \in \mathbb{R}^d$ represent the user's preferences and the item's features, respectively. Consequently, a user-item rating can be modelled as the inner product of the user embedding and the item embedding: $\hat{r}_{i,j} = \mathbf{u}_i^T \mathbf{v}_j$.

By fitting observed ratings in the training set, MF typically optimises the user and item embedding matrices by minimising the squared error loss:

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{(i,j) \in \mathcal{O}} (\hat{r}_{i,j} - r_{i,j})^2 + \lambda_U \|\mathbf{U}\|_F^2 + \lambda_V \|\mathbf{V}\|_F^2 \quad (3.1)$$

where $(i, j) \in \mathcal{O}$ indicates that the rating $r_{i,j}$ is observed, and $\lambda_U \|\mathbf{U}\|_F^2$ and $\lambda_V \|\mathbf{V}\|_F^2$ are regularisation terms to prevent overfitting.

3.2.2 Bayesian Personalised Ranking

In implicit feedback scenarios, only positive observations are available (Rendle et al., 2009). The missing data comprises a mixture of truly negative and unobserved positive feedback. Typically, positive feedback and missing data are assigned values of 1 and 0, respectively. The user-item interaction matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$ derived from implicit feedback is defined as follows:

$$s_{i,j} = \begin{cases} 1, & \text{if interaction (user } i, \text{ item } j) \text{ is observed;} \\ 0, & \text{otherwise.} \end{cases} \quad (3.2)$$

A value of 1 indicates that the user has interacted with the item; however, it does not necessarily imply that the user likes the item, as they might have accidentally clicked on it or disliked the item after purchasing and evaluating it. Similarly, a value of 0 does not necessarily mean that the user dislikes the item, as it could simply be that the user is unaware of the item's existence (He et al., 2017). This inherent noise poses a challenge in learning accurate user and item representations from implicit feedback.

The objective of ranking recommendation is to provide each user with a personalised ranked list of items. The common approach involves predicting a score $\hat{s}_{i,j}$ that indicates the interest level of user i for item j , and then sorting the items according to these scores. Although MF methods have achieved impressive results in rating prediction, they tend to perform poorly in item ranking tasks (Chen and Peng, 2018). To address this issue, Bayesian Personalised Ranking (BPR) uses item pairs as training data instead of scoring single items. The underlying idea is that a user prefers items they have interacted with over all other unobserved items. BPR-MF (Rendle et al., 2009) extends MF by learning user and item embedding matrices with a different optimisation criterion:

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{(i,j,k) \in \mathcal{D}} -\ln \sigma(\hat{s}_{i,j} - \hat{s}_{i,k}) + \lambda_U \|\mathbf{U}\|_F^2 + \lambda_V \|\mathbf{V}\|_F^2, \quad (3.3)$$

where $(i, j, k) \in \mathcal{D}$ denotes that interaction $s_{i,j}$ is observed but $s_{i,k}$ is unobserved, and $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function.

3.2.3 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) are designed to operate on graph-structured data by iteratively updating node embeddings based on their neighbours' information. At each layer, each node aggregates information from its neighbours (including itself) and updates its embedding accordingly.

Neighbour Aggregation.

In each layer, a node collects and aggregates the embeddings of its neighbours (including itself):

$$\mathbf{a}_i^{(l)} = \text{AGG} \left(\left\{ \mathbf{e}_j^{(l)} : j \in \mathcal{N}_i \cup \{i\} \right\} \right), \quad (3.4)$$

where $a_i^{(l)}$ represents the aggregated message for node i , and $\mathcal{N}_i$ denotes the set of its neighbours. The aggregation function AGG is typically a degree-normalised sum, formally given by:

$$a_i^{(l)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{1}{\sqrt{|\mathcal{N}_i| \cdot |\mathcal{N}_j|}} e_j^{(l)}, \quad (3.5)$$

where $|\mathcal{N}_i|$ and $|\mathcal{N}_j|$ denote the degrees of nodes i and j , respectively. This normalisation ensures that nodes with different numbers of neighbours contribute proportionally to the aggregation, preventing nodes with higher degrees from dominating the information flow.

Following the aggregation step, each node updates its embedding by applying a linear transformation and a non-linear activation to the aggregated message, as detailed in the next subsection.

Node Update.

The aggregated message is then transformed to produce the updated embedding:

$$e_i^{(l+1)} = \sigma(W^{(l)} a_i^{(l)} + b^{(l)}), \quad (3.6)$$

where $W^{(l)}$ and $b^{(l)}$ are the trainable weight matrix and bias, and σ is a non-linear activation function.

3.2.4 Model Architecture Motivation

When users and items are represented separately on implicit and explicit feedback graphs, message passing involves more architectural choices than in a single user-item graph. Although there are only two underlying entity types, each entity has a feedback-specific representation, resulting in four representation types: implicit users, implicit items, explicit users, and explicit items. For a target representation, information can in principle be obtained from three sources: (1) neighbouring nodes on the same graph, (2) the corresponding representation of the same entity on the other graph, and (3) neighbouring nodes on the other graph.

Figure 3.1 illustrates these possibilities using the implicit item z_j as the target. It may receive information from neighbouring implicit users through conventional intra-graph propagation, from its explicit counterpart v_j , or from explicit users connected to

the same item. Similar candidate pathways can be defined for the other three feedback-specific representations. The architectural problem is therefore not simply whether information should be exchanged across the two graphs, but which feedback-specific representations should exchange information and how that transferred information should subsequently propagate through the collaborative neighbourhoods.

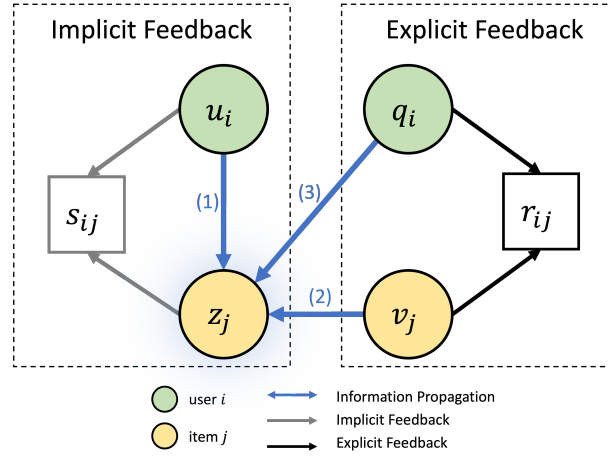


FIGURE 3.1: An illustration of three possible sources of information for message aggregation in a dual-graph setting. The implicit and explicit representations of user i and item j are denoted by implicit user u_i , implicit item z_j , explicit user q_i , and explicit item v_j , respectively. Taking the implicit item z_j as an example, its candidate information sources include (1) neighbouring implicit users on the same graph, (2) its explicit counterpart v_j , and (3) neighbouring explicit users on the other graph. The figure illustrates the architectural choices that arise when learning feedback-specific representations across the implicit and explicit graphs; not all candidate pathways are necessarily used in the final IEL architecture.

The order of inter-graph and intra-graph propagation also affects how cross-feedback information is distributed. If inter-graph transfer is performed before intra-graph propagation, information received from the other feedback type can subsequently be propagated through the collaborative neighbourhood of the target graph. For example, after an implicit item representation receives information from its explicit user neighbours, this transferred information can contribute to later updates of neighbouring implicit users and higher-order implicit nodes. By contrast, performing intra-graph propagation first would restrict the cross-feedback information introduced at that stage from influencing the earlier within-graph updates.

IEL therefore performed the selected inter-graph transfer before the corresponding intra-graph propagation. This ordering reflected the modelling objective that information originating from one feedback type should not only modify the directly connected representation but should also be able to influence its collaborative neighbourhood on the other graph. The experiments in this study did not exhaustively compare all possible propagation orders, so this design should be interpreted as a motivated architectural choice rather than an empirically established optimum.

To keep the cross-graph architecture tractable, IEL selected only a subset of the possible inter-graph pathways. The selection was guided by a modelling hypothesis about how behavioural engagement and explicit evaluation could provide complementary information. In the implicit-feedback view, an observed interaction indicated that a user had interacted with an item but discarded the numerical rating associated with that interaction. By contrast, the explicit-feedback view preserved the rating value and therefore contained information about the strength and direction of the user's evaluation. We therefore hypothesised that an implicit item representation could benefit from information propagated from its explicit user neighbours. This pathway allowed evaluative information associated with the item to supplement its representation on the implicit-feedback graph.

The reverse direction was motivated by earlier unified recommendation models. SVD++ and related neighbourhood-based approaches incorporated representations of implicitly interacted items into an explicit user representation, showing that a user's behavioural history can complement the representation used for explicit preference prediction (Bell and Koren, 2007; Koren, 2008a; Liu et al., 2017). IEL retained this principle by allowing the explicit user representation to receive information from its implicit item neighbours. The two selected pathways therefore formed a complementary pair: explicit user information enriched the implicit representation of an item, while implicit item information enriched the explicit representation of a user.

These bridge representations were selected because they provided a simple way to connect behavioural and evaluative information without introducing every possible cross-graph connection. After the inter-graph update, intra-graph message passing allowed the transferred information to propagate through the corresponding collaborative neighbourhood. This design operationalised the central hypothesis of IEL: feedback-specific representations should remain distinct, but selected representations should exchange complementary information through their neighbours. Alternative bridge nodes, bidirectional counterpart connections, or learnable cross-graph routing

mechanisms were not exhaustively evaluated and remain possible extensions of the architecture.

3.3 Interdependent Embedding Learning

We developed a GCN-based recommendation framework named IEL to model the process through which users interact with items implicitly and explicitly, based on three key designs:

1. The user/item in implicit feedback and the user/item in explicit feedback can be characterised by two different embeddings,
2. the implicit item embedding depends on the explicit embeddings of its neighbouring users, and
3. the explicit user embedding depends on the implicit embeddings of their neighbouring items.

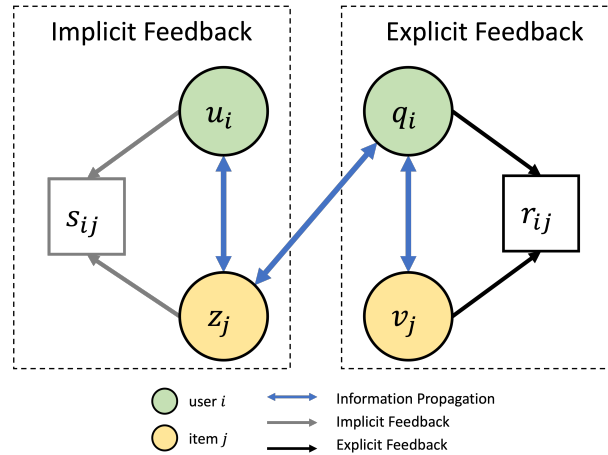


FIGURE 3.2: The overall framework of IEL, a GCN-based recommendation model designed to integrate implicit and explicit feedback. Each user and item is represented by two distinct embeddings for the two feedback types: u_i and q_i for user i , and z_j and v_j for item j . Implicit and explicit feedback are modelled via message passing within and across graphs. Blue arrows denote the direction of message aggregation. Specifically, z_j integrates information from explicit user embeddings, while q_i aggregates messages from implicit item embeddings, enabling inter-graph information transfer.

An overview of our approach is provided in Figure 3.2, and the notations used are summarised in Table 3.2. In the framework, nodes and arrows denote variables and their relationships, respectively. There are six variables: two represent implicit feedback (s_{ij}) and explicit feedback (r_{ij}), while the remaining four represent user i 's implicit embedding (u_i), user i 's explicit embedding (q_i), item j 's implicit embedding (z_j), and item j 's explicit embedding (v_j). The implicit and explicit embeddings of users and items determine the implicit and explicit feedback, respectively. The blue arrows indicate the source and target of message passing. In our design, users and items under implicit feedback absorbed information from their neighbours to update their own embeddings, and a similar process occurred for explicit feedback. There was also message passing between different feedback: the item's implicit embedding and the user's explicit embedding acted as "bridges" to facilitate information propagation between feedback. In the following sections, we introduce our intra-feedback GCN and inter-feedback GCN.

TABLE 3.2: Summary of notations.

Symbol	Description
m	number of users
n	number of items
i	$i \in \{1, \dots, m\}$ is used to index a user
j	$j \in \{1, \dots, n\}$ is used to index an item
s_{ij}	the implicit feedback of user i for item j
r_{ij}	the explicit feedback of user i for item j
$u_i \in \mathbb{R}^d$	user i 's embedding in implicit feedback
$q_i \in \mathbb{R}^d$	user i 's embedding in explicit feedback
$z_j \in \mathbb{R}^d$	item j 's embedding in implicit feedback
$v_j \in \mathbb{R}^d$	item j 's embedding in explicit feedback
$\mathcal{S}_i$	the item set that user i has implicitly interacted with
$\mathcal{R}_i$	the item set that user i has explicitly interacted with
$\mathcal{S}_j$	the user set that item j has implicitly interacted with
$\mathcal{R}_j$	the user set that item j has explicitly interacted with

3.3.1 Intra-Feedback GCN

For the propagation rule on the implicit graph, we followed the straightforward yet highly effective design of the LightGCN model (He et al., 2020), which has demonstrated superior performance on ranking tasks.

$$\mathbf{u}_i^{(l+1)} = \frac{1}{\sqrt{|\mathcal{S}_i|}} \sum_{j \in \mathcal{S}_i} \frac{1}{\sqrt{|\mathcal{S}_j|}} \mathbf{z}_j^{(l)}, \quad \mathbf{z}_j^{(l+1)} = \frac{1}{\sqrt{|\mathcal{S}_j|}} \sum_{i \in \mathcal{S}_j} \frac{1}{\sqrt{|\mathcal{S}_i|}} \mathbf{u}_i^{(l)}, \quad (3.7)$$

where $\mathbf{u}_i^{(l+1)}$ denotes user i 's implicit embedding at layer $l + 1$; $\mathbf{z}_j^{(l)}$ denotes item j 's implicit embedding at layer l ; $\mathcal{S}_i$ and $\mathcal{S}_j$ are the sets of user i 's and item j 's immediate (first-order) neighbours under implicit feedback, respectively; $|\mathcal{S}|$ represents the number of elements in a set $\mathcal{S}$.

It can be observed that, to prevent the scale of embeddings from increasing with graph convolution operations (Kipf and Welling, 2016), the method involves applying symmetric square root normalisation to each neighbour's embedding during neighbourhood aggregation. Specifically, this normalisation is applied to both the target node and the neighbouring nodes. The purpose is to emphasise the importance of information from neighbours that have fewer interactions in the dataset. After aggregating the weighted information from all the neighbouring nodes, a simple weighting is applied based on the target node's own number of interactions, which is essentially the number of its neighbours.

On the explicit graph, we adopted a similar mechanism to aggregate features from the target node's neighbours:

$$\mathbf{q}_i^{(l+1)} = \frac{1}{\sqrt{|\mathcal{R}_i|}} \sum_{j \in \mathcal{R}_i} \frac{1}{\sqrt{|\mathcal{R}_j|}} \mathbf{v}_j^{(l)}, \quad \mathbf{v}_j^{(l+1)} = \frac{1}{\sqrt{|\mathcal{R}_j|}} \sum_{i \in \mathcal{R}_j} \frac{1}{\sqrt{|\mathcal{R}_i|}} \mathbf{q}_i^{(l)}, \quad (3.8)$$

where $\mathbf{q}_i^{(l+1)}$ denotes user i 's explicit embedding at layer $l + 1$; $\mathbf{v}_j^{(l)}$ denotes item j 's explicit embedding at layer l ; $\mathcal{R}_i$ and $\mathcal{R}_j$ are the sets of user i 's and item j 's direct neighbours under explicit feedback; $|\mathcal{R}|$ represents the number of elements in a set $\mathcal{R}$.

3.3.2 Inter-Feedback GCN

The key point is to exchange information between the two graphs. Before passing message within the same graph, nodes first collect features from their neighbours on the other graph. In this process, there are four possible options: the implicit user embedding can absorb information from their neighbouring items on the explicit graph ($\mathbf{u}_i^{(l+1)} = \text{AGG}(\mathbf{v}_j^{(l)} : j \in \mathcal{R}_i)$), and vice versa ($\mathbf{v}_j^{(l+1)} = \text{AGG}(\mathbf{u}_i^{(l)} : i \in \mathcal{S}_j)$); similarly, the implicit item embedding can absorb information from their neighbouring users on the explicit graph ($\mathbf{z}_j^{(l+1)} = \text{AGG}(\mathbf{q}_i^{(l)} : i \in \mathcal{R}_j)$), and vice versa ($\mathbf{q}_i^{(l+1)} = \text{AGG}(\mathbf{z}_j^{(l)} :$

$j \in \mathcal{S}_i$)). As shown in the framework diagram (Figure 3.2), we chose implicit items and explicit users as representatives for inter-feedback message passing:

$$\mathbf{z}_j^{(l)} = \mathbf{z}_j^{(l)} + \frac{1}{\sqrt{|\mathcal{R}_j|}} \sum_{i \in \mathcal{R}_j} \mathbf{q}_i^{(l)}, \quad (3.9)$$

$$\mathbf{q}_i^{(l)} = \mathbf{q}_i^{(l)} + \frac{1}{\sqrt{|\mathcal{S}_i|}} \sum_{j \in \mathcal{S}_i} \mathbf{z}_j^{(l)}. \quad (3.10)$$

The reasons for this choice, beyond the intuition mentioned in Introduction, included our aim to avoid making the message aggregation between graphs overly complex, which could potentially impact model performance. Furthermore, we advocated that as long as the order of information propagation within and between feedback is clarified, nodes that do not directly participate in inter-feedback message aggregation can still access information from nodes in the other feedback.

Specifically, we first performed message passing between feedback, followed by within feedback. Taking a single-layer GCN as an example, the implicit item embedding at layer 0 first absorbs the information of its neighbours in the explicit feedback, i.e., the explicit user embeddings ($\mathbf{q}_i^{(0)}$). The gathered neighbour information is then added to the item's own embedding ($\mathbf{z}_j^{(0)}$). We referred to this integration of neighbouring information from the other feedback as implicit item inter-embedding at layer 0 ($inter\text{-}\mathbf{z}_j^{(0)}$). Correspondingly, the explicit user embedding at layer 0 also obtains its inter-embedding ($inter\text{-}\mathbf{q}_i^{(0)}$):

$$inter\text{-}\mathbf{z}_j^{(0)} = \mathbf{z}_j^{(0)} + \frac{1}{\sqrt{|\mathcal{R}_j|}} \sum_{i \in \mathcal{R}_j} \mathbf{q}_i^{(0)}, \quad inter\text{-}\mathbf{q}_i^{(0)} = \mathbf{q}_i^{(0)} + \frac{1}{\sqrt{|\mathcal{S}_i|}} \sum_{j \in \mathcal{S}_i} \mathbf{z}_j^{(0)}. \quad (3.11)$$

Next, this inter-embedding undergoes message passing within the respective feedback:

$$\mathbf{u}_i^{(1)} = \frac{1}{\sqrt{|\mathcal{S}_i|}} \sum_{j \in \mathcal{S}_i} \frac{1}{\sqrt{|\mathcal{S}_j|}} inter\text{-}\mathbf{z}_j^{(0)}, \quad \mathbf{z}_j^{(1)} = \frac{1}{\sqrt{|\mathcal{S}_j|}} \sum_{i \in \mathcal{S}_j} \frac{1}{\sqrt{|\mathcal{S}_i|}} \mathbf{u}_i^{(0)}, \quad (3.12)$$

$$\mathbf{q}_i^{(1)} = \frac{1}{\sqrt{|\mathcal{R}_i|}} \sum_{j \in \mathcal{R}_i} \frac{1}{\sqrt{|\mathcal{R}_j|}} \mathbf{v}_j^{(0)}, \quad \mathbf{v}_j^{(1)} = \frac{1}{\sqrt{|\mathcal{R}_j|}} \sum_{i \in \mathcal{R}_j} \frac{1}{\sqrt{|\mathcal{R}_i|}} inter\text{-}\mathbf{q}_i^{(0)}. \quad (3.13)$$

Equation 3.12 implies that the implicit user embedding at layer 1 ($\mathbf{u}_i^{(1)}$) not only receives the implicit embeddings of its neighbouring items from layer 0 but also the explicit information from layer 0. This realises our concept that even nodes not directly participating in inter-feedback message aggregation can still access information from the other feedback. The implicit item embedding at layer 1 ($\mathbf{z}_j^{(1)}$) includes the implicit information of its neighbouring users from layer 0. Equation 3.13 carries the same implication.

It is worth noting that in the inter-feedback GCN, we did not use the symmetric square root normalisation of the intra-feedback GCN. Specifically, we only applied normalisation to the target node, not to the neighbouring nodes. Our reasoning was that the process of the implicit item embedding aggregating the explicit user neighbour information corresponds to users viewing the explicit feedback of items (such as reviews): users are primarily concerned with the number of explicit feedback received by the current item, rather than the total feedback given by those users. In other words, each neighbour information is equally important in this process, regardless of the overall feedback activity of the neighbours. A similar rationale applied to the explicit user embedding aggregating its implicit item neighbour information. Moreover, this design (Equation 3.10) is similar to the structure of SVD++ (Koren, 2008a), where only the feedback quantity of the target node is considered when aggregating neighbour information between feedback.

3.3.3 Layer Combination and Model Prediction

Layer Combination

After obtaining the representations from each layer, we used a self-attention mechanism to integrate the information from each layer to produce the final representation:

$$\begin{aligned}
 \mathbf{u}_i &= \sum_{l=0}^L \alpha_{u,l} \mathbf{u}_i^{(l)}, \\
 \mathbf{z}_j &= \sum_{l=0}^L \alpha_{z,l} \text{inter-}\mathbf{z}_j^{(l)}, \\
 \mathbf{q}_i &= \sum_{l=0}^L \alpha_{q,l} \text{inter-}\mathbf{q}_i^{(l)}, \\
 \mathbf{v}_j &= \sum_{l=0}^L \alpha_{v,l} \mathbf{v}_j^{(l)}.
 \end{aligned} \tag{3.14}$$

Taking the implicit user embedding $\mathbf{u}_i$ as an example, $\alpha_{u,l}$ is computed as follows:

$$\alpha'_{u,l} = \alpha_u \text{LeakyReLU}(\mathbf{W}_u \mathbf{u}_i^{(l)}), \quad \alpha_{u,l} = \frac{\exp(\alpha'_{u,l})}{\exp(\sum_{l'=0}^L \alpha'_{u,l'})}. \quad (3.15)$$

where $\mathbf{W}_u \in \mathbf{R}^{d \times d}$ denotes the linear transformation; $\text{LeakyReLU}(\cdot)$ is an activation function; and $\alpha_u \in \mathbf{R}^{d \times 1}$ is a context vector.

Model Prediction

In this work, we chose the inner product to estimate the user's implicit score towards the target item:

$$\hat{s}_{i,j} = \mathbf{u}_i^T \mathbf{z}_j \quad (3.16)$$

For the explicit score, we applied a non-linear transformation to the explicit user embedding and item embedding respectively before computing their inner product:

$$\bar{\mathbf{q}}_i = \text{LeakyReLU}(\mathbf{W}_1 \mathbf{q}_i), \quad \bar{\mathbf{v}}'_j = \text{LeakyReLU}(\mathbf{W}_2 \mathbf{v}_j). \quad (3.17)$$

$$\hat{r}_{i,j} = \text{sigmoid}(\bar{\mathbf{q}}_i^T \bar{\mathbf{z}}_j), \quad (3.18)$$

where the sigmoid operation in $\hat{r}_{i,j}$ is to bound the prediction range to $[0, 1]$. Correspondingly, we mapped the raw rating $r_{i,j}$ to an interval $[0, 1]$ by employing the function $y(x) = (x - r_{\min}) / (r_{\max} - r_{\min})$, where $r_{\max}$ and $r_{\min}$ are the maximum and minimum rating in the dataset, respectively. For example, $r_{\max}$ equals to 5 and $r_{\min}$ equals to 1 in many websites (e.g., Amazon, Yelp) that use a five-star rating system.

3.3.4 Model Optimisation

Since our model aimed to accomplish both item ranking and rating prediction tasks, it fell under multi-task learning (MTL). Most MML approaches use a combined loss function with task weights to control the importance of each task. However, tuning the weights of multiple task losses is time-consuming, and fixed weights do not apply to all batches throughout the dynamic training process (Xu et al., 2023). Therefore, we adopted a step-wise training approach. The advantage of this method is that we do not need to adjust the hyperparameters for task weights.

Specifically, if the target task was item ranking, we first performed preliminary batch training on the rating prediction task based on Equation 3.1, updating the model parameters related to rating prediction in one batch. We then train the ranking task based on Equation 3.2 and update its corresponding parameters on the same batch. Conversely, if rating prediction was the target task, we conducted preliminary batch training on the ranking task before training the rating prediction task. This ensured that the parameters updated through target task learning were used during model prediction.

3.3.5 Design Decisions and Other Architectural Choices

In this section, we describe some of the design choices we considered for the IEL architecture and explain why they did not yield satisfactory results in our preliminary experiments.

Using Attention Mechanism Instead of Linear Message Passing

In order to capture the varying contributions of each neighbour to the target node, we adopted the attention mechanisms proposed in (Brody et al., 2021) and (Vaswani et al., 2017) for intra-graph and inter-graph message passing, respectively. However, the item ranking performance did not match the results achieved with the current design as outlined in (He et al., 2020), although the rating prediction results were comparable.

Adopting Non-Linear Prediction for Implicit Scores

We considered applying a non-linear transformation to the implicit user embedding and item embedding respectively before computing their inner product to estimate the implicit score, similar to the method used for explicit user/item embeddings before calculating the explicit score. Alternatively, we could concatenate the implicit user embedding and item embedding and pass them through a three-layer non-linear network to obtain the implicit score. However, we found that such approaches only degraded the ranking quality. In contrast, the simple inner product yielded the best results.

Node Learning versus Edge Learning

In our initial experimental design, we utilised edge learning instead of node learning. Specifically, we assigned an embedding to each edge and used the aforementioned attention mechanisms to update the edge representations. Finally, the predicted implicit and explicit edges were passed through a non-linear network to obtain the implicit and explicit scores, respectively. However, this approach resulted in suboptimal ranking quality, despite achieving comparable results for rating prediction. Another reason we abandoned edge learning in this model was computational resource constraints. In recommendation systems, the number of edges grows with the number of users and items, making edge learning significantly more resource-intensive than node learning. Additionally, edge learning was incapable of predicting isolated users or items (those with no interactions in the train dataset). Even when we attempted to assign initial values to all nodes (including isolated nodes) and update them along with other parameters, enabling predictions based on the node’s own information, the ranking quality did not improve.

3.4 Experiments

We conducted experiments on two real-world datasets: Amazon-Office and Amazon-Video Games. We compared our method against four competitive models. For the item ranking task, we employed Recall@k, Precision@k, Normalised Discounted Cumulative Gain (NDCG@k), and naive-NDCG@k metrics to evaluate model performance, where $k \in \{3, 5, 10, 20\}$. For the rating prediction task, we utilised mean square error (MSE) and mean absolute error (MAE) as evaluation metrics. The experimental results demonstrated that IEL outperforms the competing models in both tasks. Additionally, we examined the impact of the number of GCN layers on the model’s performance.

3.4.1 Experimental Setup

TABLE 3.3: Statistics of the datasets.

Dataset	Users	Items	Interactions	Density
Amazon-Office	3,613	2,427	58,300	0.00665
Amazon-Video Games	5,942	3,793	103,778	0.00460

Datasets To assess the effectiveness of IEL, we conducted experiments on two Amazon product-category datasets (Ni et al., 2019): Office and Video Games. These datasets, which are publicly accessible, differ in domain, size, and density. The statistics for the two datasets are summarised in Table 3.3. The Amazon review¹ dataset is a commonly utilised dataset for product recommendation. From this collection, we selected the Office and Video Games subsets. Ratings in the dataset are provided on a five-star scale. To ensure the quality of the dataset, we used the 10-core setting, retaining only users and items with at least ten interactions.

For each dataset, the historical interactions of each user were split into a training set (70%), a validation set (10%), and a test set (20%) based on chronological order. All models were trained on the training dataset, hyperparameters were tuned on the validation dataset, and performance was reported on the test dataset. Specifically, we selected the model that achieved the best performance on the validation dataset and evaluated it on the test dataset. Following the treatment of observed interactions as implicit signals in previous recommendation studies (Koren, 2008a; He et al., 2015), we constructed the implicit-feedback view from the occurrence of observed ratings. Specifically, each observed user-item rating was assigned a binary interaction value of 1 in the implicit graph, while the corresponding numerical rating was retained in the explicit graph. Consequently, the implicit and explicit graphs had the same observed user-item connectivity in these experiments, but represented different information: the implicit graph encoded whether an interaction occurred, whereas the explicit graph additionally preserved the strength of the user’s evaluation.

Baselines To demonstrate the effectiveness of our proposed IEL model, we compared it against the following methods²:

- **BPR-MF** (Rendle et al., 2009): BPR-MF is a matrix factorisation approach optimised using BPR loss, which leverages pair-wise ranking between observed and missing interactions. It utilises implicit feedback exclusively.
- **SVD++** (Koren, 2008a): This algorithm generates the explicit user embedding by considering all items for which the user provided implicit feedback. It employs

¹<https://nijianmo.github.io/amazon/index.html>

²The bias terms (i.e., user bias, item bias, and global bias) have been shown to significantly enhance rating prediction performance (Sachdeva and McAuley, 2020). To ensure a fair comparison, specifically to evaluate the effectiveness of the model architecture itself, we omitted the bias components in certain models that originally include them, such as SVD++.

both implicit and explicit feedback but is utilised solely for the rating prediction task.

- **PMF+P+N** (Chen and Peng, 2018): PMF+P+N is a matrix co-factorisation algorithm that simultaneously utilises implicit and explicit feedback. In this model, there is no relationship between the implicit and explicit item embeddings, and users share the same embedding in both implicit and explicit feedback.
- **RnR** (Hadash et al., 2018): RnR is a unified recommendation framework where the explicit user and item embedding are derived from their respective implicit embedding. For a fair comparison, we adhered to its original setting, adopting BPR-MF (Rendle et al., 2009) as the ranking model and SVD (Paterek, 2007) as the rating prediction model. It is noteworthy that RnR did not report their rating prediction quality. However, we found that their model can achieve superior performance in the rating prediction task. Therefore, in this study, we also included its evaluation results on this task.
- **LightGCN** (He et al., 2020) is a GCN-based ranking recommendation model that retains only its most essential component, i.e., neighbourhood aggregation. It has been widely adopted as a strong GCN-based baseline for collaborative filtering. Given that it formed the foundation of our model, we included it as one of the baselines for the ranking task.

Evaluation Protocols In the ranking task, for each user in the validation/test dataset, we predicted their implicit score for all items they had not interacted with in the training dataset. These items were then ranked according to their predicted values. To evaluate the performance of all models for item ranking, we used Recall@k, Precision@k, naive-NDCG@k, and NDCG@k as evaluation metrics, where $k \in \{3, 5, 10, 20\}$. The formulas for these metrics are presented below.

$$\text{Precision@k} = \frac{\sum_{i=1}^k I(i)}{k}, \quad (3.19)$$

where $I(i)$ is an indicator function that equals 1 if the i -th recommended item has been interacted with by the user in the validation/test dataset, and 0 otherwise.

$$\text{Recall@k} = \frac{\sum_{i=1}^k I(i)}{|\mathcal{G}|}, \quad (3.20)$$

where $|\mathcal{G}|$ denotes the total number of items the user has interacted with in the validation/test dataset.

$$\begin{aligned} \text{NDCG@k} &= \frac{\text{DCG@k}}{\text{IDCG@k}}, \\ \text{DCG@k} &= \sum_{i=1}^k \frac{2^{\text{rel}(i)} - 1}{\log_2(i+1)}, \end{aligned} \quad (3.21)$$

where $\text{rel}(i)$ is the predicted implicit score of the i -th recommended item; IDCG@k , defined as the "Ideal DCG", is calculated by sorting the relevance scores in descending order under the ideal scenario (ground truth). We also reported the naive-NDCG@k value, which is computed by replacing $\text{rel}(i)$ with $I(i)$ in the NDCG@k formula. The final results were averaged across all users in the validation/test data.

In the rating prediction task, we calculated the MSE and MAE between all actual ratings and the predicted explicit ratings in the validation/test dataset:

$$\text{MSE} = \frac{\sum_{(i,j) \in \mathcal{T}} (\hat{r}_{i,j} - r_{i,j})^2}{|\mathcal{T}|}, \quad (3.22)$$

$$\text{MAE} = \frac{\sum_{(i,j) \in \mathcal{T}} |\hat{r}_{i,j} - r_{i,j}|}{|\mathcal{T}|}, \quad (3.23)$$

where $\mathcal{T}$ is the validate/test dataset.

Parameter Settings We implemented our IEL model using PyTorch. For all models, the embedding size was fixed at 64 and the batch size was set to 1024. A grid search was conducted for hyperparameter tuning: the learning rate was varied among $\{1e-5, 1e-4, 1e-3, 1e-2\}$, and the coefficient for L2 regularisation was searched within $\{1e-6, 1e-5, 1e-4, 1e-3\}$. Additionally, an early stopping strategy was employed, halting training if the monitored metric NDCG@20 on the validation data did not improve for 30 consecutive epochs in the ranking task, or if the MSE did not decrease for 30 epochs in the rating prediction task. Regarding the number of negative samples in the BPR loss, we randomly sampled one unobserved interaction for each user. For PMF+P+N and RnR, the range of the implicit task weight α was set to $[0.1, 1.0]$ with a step size of 0.1.

3.4.2 Item Ranking Performance

Table 3.4 shows the item ranking performance of all models. We summarise the following observations:

- LightGCN proved to be the strongest baseline, outperforming all other baselines on most metrics across both datasets. The only exception was on the Amazon Office dataset, where its (naive-)NDCG@3 to @5 scores were slightly lower than those of PMF+P+N.
- PMF+P+N exhibited the most variability in performance. Specifically, it showed robust results on the Amazon Office dataset, achieving approximately a 20% improvement over BPR-MF and RnR on naive-NDCG@3 and NDCG@3 metrics. Although its advantage diminished with increasing recommendation list length, it still outperformed them by about 7% on (naive-)NDCG@20. However, PMF+P+N's performance on the Amazon-Video Games dataset was subpar, even lagging behind BPR-MF. We hypothesised that this was due to the presence of items in the Video Games validation and test datasets that were not seen during training, which PMF+P+N struggled to predict accurately.
- In contrast, RnR demonstrated more consistent results. It surpassed PMF+P+N on Video Games, becoming the second strongest baseline for this dataset.
- Overall, BPR-MF, as a single model, underperformed compared to other unified models on both datasets. Interestingly, on the Video Games dataset, BPR-MF excelled when the recommendation list length was 20, even surpassing the high-performing RnR.
- IEL consistently outperformed all baselines on both datasets. Its improvements on various metrics were generally around 10% or higher, except for Recall@10 and Precision@10 on Office, where the smallest improvement was still approximately 5%. This indicated that our model achieved significantly better results in the item ranking task compared to other single and unified models. Notably, IEL's enhancements were more pronounced on the Office dataset overall.
- When compared to the strongest baseline, IEL steadily achieved greater improvements in the NDCG metric than naive-NDCG, except for the special cases where PMF+P+N served as the strongest baseline on the Office dataset (@3 and @5).

For instance, on the Video Games dataset, IEL improved NDCG@20 by 9.41%, surpassing the 9.10% improvement of naive-NDCG@20. The former accounts for multiple rating values, whereas the latter only considers binary values. This suggests that our model prioritises ranking items with higher user ratings towards the top, rather than merely considering whether a user interacts with an item, even if they subsequently give it a lower rating.

TABLE 3.4: Overall comparison of ranking performance. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Amazon-Office				Amazon-Video Games			
Method	Recall@3	Precision@3	n-ndcg@3	ndcg@3	Recall@3	Precision@3	n-ndcg@3	ndcg@3
BPR-MF	0.04551	0.05240	0.10132	0.10060	0.01396	0.01683	0.03171	0.03126
PMF+P+N	0.05266	0.06135	<u>0.12289</u>	<u>0.12240</u>	0.01318	0.01638	0.03049	0.02999
RnR	0.04625	0.05314	0.10240	0.10208	0.01443	0.01767	0.03176	0.03113
LightGCN	<u>0.05356</u>	<u>0.06191</u>	0.11473	0.11401	<u>0.01552</u>	<u>0.01919</u>	<u>0.03532</u>	<u>0.03485</u>
IEL	0.06337	0.07298	0.13704	0.13623	0.01774	0.02160	0.03800	0.03790
%Improv.	18.32%	17.88%	11.51%	11.30%	14.30%	12.56%	7.59%	8.75%
Method	Recall@5	Precision@5	n-ndcg@5	ndcg@5	Recall@5	Precision@5	n-ndcg@5	ndcg@5
BPR-MF	0.06164	0.04323	0.12002	0.11928	0.02104	0.01565	0.04139	0.04098
PMF+P+N	0.06715	0.04744	<u>0.13950</u>	<u>0.13894</u>	0.01988	0.01525	0.03959	0.03897
RnR	0.06088	0.04268	0.11920	0.11864	0.02200	0.01619	0.04159	0.04086
LightGCN	<u>0.07381</u>	<u>0.05159</u>	0.13733	0.13655	<u>0.02383</u>	<u>0.01757</u>	<u>0.04545</u>	<u>0.04488</u>
IEL	0.08057	0.05679	0.15650	0.15549	0.02706	0.01949	0.04897	0.04872
%Improv.	9.16%	10.08%	12.19%	11.91%	13.55%	10.93%	7.74%	8.56%
Method	Recall@10	Precision@10	n-ndcg@10	ndcg@10	Recall@10	Precision@10	n-ndcg@10	ndcg@10
BPR-MF	0.09000	0.03186	0.14649	0.14568	0.03657	0.01358	0.05706	0.05647
PMF+P+N	0.09035	0.03266	0.16122	0.16045	0.03485	0.01331	0.05502	0.05426
RnR	0.09082	0.03236	0.14751	0.14694	0.03787	0.01402	0.05772	0.05678
LightGCN	<u>0.10664</u>	<u>0.03806</u>	<u>0.16781</u>	<u>0.16654</u>	<u>0.04212</u>	<u>0.01555</u>	<u>0.06333</u>	<u>0.06254</u>
IEL	0.11204	0.04052	0.18692	0.18572	0.04722	0.01750	0.06961	0.06911
%Improv.	5.06%	6.46%	11.39%	11.52%	12.11%	12.54%	9.92%	10.50%
Method	Recall@20	Precision@20	n-ndcg@20	ndcg@20	Recall@20	Precision@20	n-ndcg@20	ndcg@20
BPR-MF	0.12614	0.02306	0.17485	0.17363	0.06565	0.01234	0.08067	0.07983
PMF+P+N	0.12347	0.02294	0.18766	0.18673	0.06035	0.01159	0.07649	0.07560
RnR	0.12735	0.02324	0.17572	0.17482	0.06467	0.01221	0.07958	0.07855
LightGCN	<u>0.14680</u>	<u>0.02696</u>	<u>0.19844</u>	<u>0.19698</u>	<u>0.07239</u>	<u>0.01351</u>	<u>0.08779</u>	<u>0.08679</u>
IEL	0.15942	0.02956	0.22275	0.22122	0.08120	0.01495	0.09577	0.09496
%Improv.	8.60%	9.64%	12.25%	12.30%	12.17%	10.66%	9.10%	9.41%

3.4.3 Rating Prediction Performance

Table 3.5 summarises all rating prediction results of all models. Compared to its robust performance in item ranking, PMF+P+N demonstrated weaker results in the rating

prediction task. However, as a unified model, it still outperformed the classical single model SVD++. Although SVD++ also utilised implicit item embeddings, it did not employ a ranking algorithm for training. PMF+P+N achieved an improvement of approximately 8% over SVD++ (except for a 3% improvement in MAE on the Office dataset).

RnR can be considered the strongest baseline for rating prediction. It improved SVD++ by around 10% on both metrics on the Office dataset and by 20% on the Video Games dataset. Compared to PMF+P+N, RnR’s MSE and MAE on the Office dataset exceeded PMF+P+N by approximately 3% and 8%, respectively, and by 12% on both metrics on the Video Games dataset. This explains why, despite RnR not reporting rating prediction performance in their original paper, we included its performance in this task in our evaluation. We hypothesised that RnR’s superior performance in rating prediction was due to its explicit user/item embeddings being non-linearly transformed from their respective implicit embeddings, thereby incorporating more information for rating prediction. This rationale also underlay our similar approach to the final explicit embeddings in this study.

IEL consistently yielded the best performance across both datasets, except for MAE on Office. Although it did not exhibit a substantial advantage over RnR, it significantly outperformed SVD++ and PMF+P+N.

TABLE 3.5: Overall comparison of rating prediction performance. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Amazon-Office		Amazon-Video Games	
Method	MSE	MAE	MSE	MAE
SVD++	0.04044	0.14185	0.07936	0.23000
PMF+P+N	0.03703	0.13807	0.07349	0.20705
RnR	<u>0.03587</u>	0.12747	<u>0.06434</u>	<u>0.18171</u>
IEL	0.03570	<u>0.12773</u>	0.06266	0.18107
%Improv.	0.47%	-0.20%	2.61%	0.35%

3.4.4 Impact of Number of Layers

To investigate whether IEL can benefit from multiple message aggregation layers, we varied the model depth on both tasks. Specifically, we searched the number of layers within the range of {1, 2, 3, 4}.

Item Ranking

Figures 3.3 and 3.4 show the ranking metrics@20 for different layer numbers on the Office and Video Games datasets, respectively.

On the Office dataset, we observed that the NDCG metrics achieved their highest values at three layers, while Recall and Precision were optimal at either two or four layers. This indicates that the model’s performance with multiple layers surpasses that with a single layer, suggesting that considering only first-order neighbours is insufficient. High-order neighbour information is beneficial for embedding learning and enhances model performance, consistent with previous GCN studies. Additionally, the NDCG metrics showed a decline at four layers, possibly indicating the onset of overfitting.

In contrast, IEL exhibited more consistent performance on the Video Games dataset, with all metrics reaching their peak at four GCN layers. This suggests that there is potential for further increasing model depth on this dataset, though exploring GCN depth is beyond the scope of this study and is left for future work. We hypothesised that the collaborative signal in the Video Games dataset is stronger, meaning that the incorporation of neighbour signals is more advantageous for learning node embeddings.

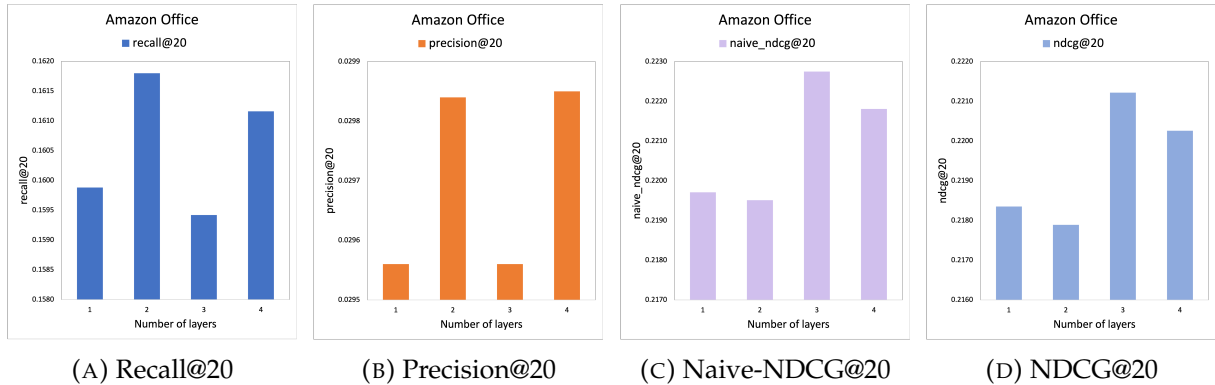


FIGURE 3.3: Ranking performance of IEL with varying GCN layers on the **Amazon-Office** dataset.

Rating Prediction

Figure 3.5 illustrated the MSE and MAE values for varying GCN layer numbers across the two datasets. We observed that, similar to the item ranking task, a single GCN layer, which considers only the first-order neighbours of the target node, did not yield

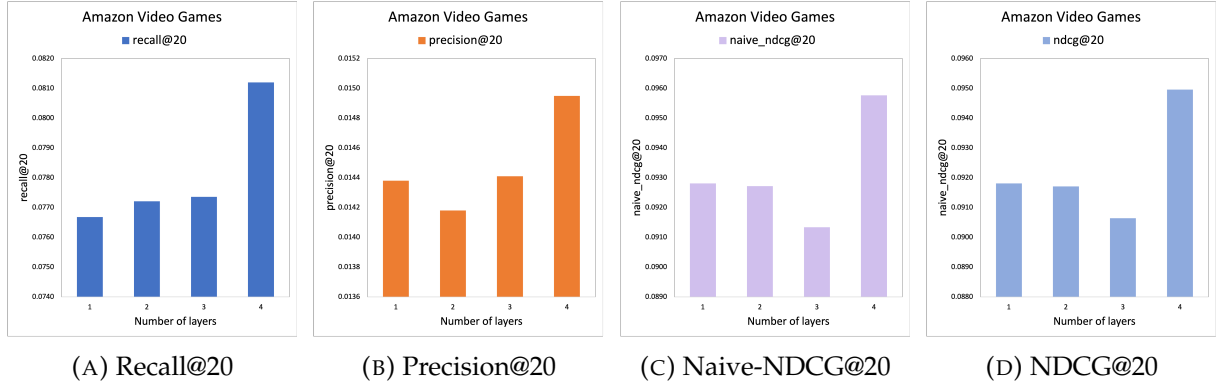


FIGURE 3.4: Ranking performance of IEL with varying GCN layers on the **Amazon-Video Games** dataset.

optimal performance for the rating prediction task. This was evidenced by the experimental results, which generally indicated that the prediction error was minimised with three layers on both datasets. Compared to MSE, the trend for MAE was more pronounced, showing a clear decrease followed by an increase as the number of layers increased.

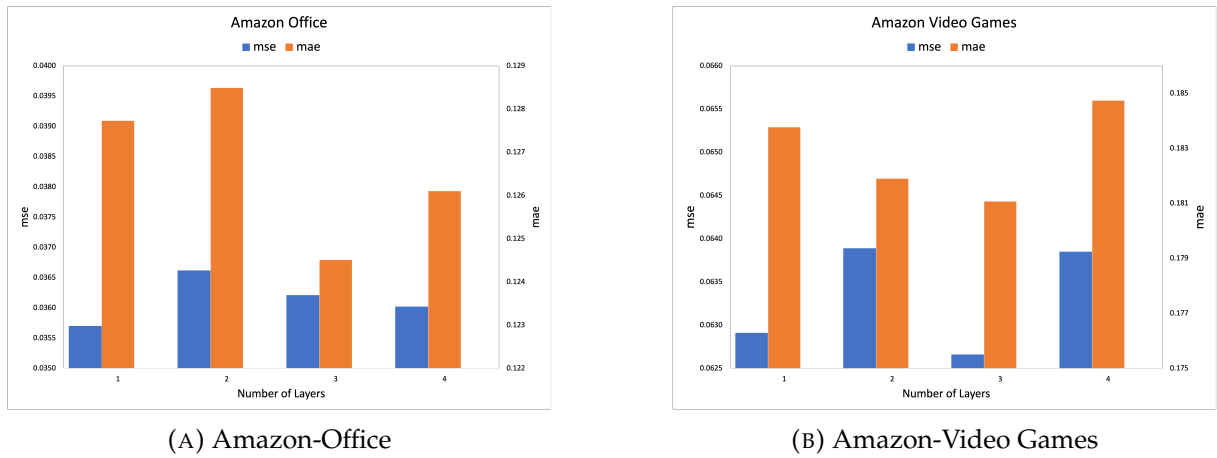


FIGURE 3.5: Rating prediction performance of IEL with varying GCN layers across different datasets.

3.5 Discussion

Implicit and explicit feedback differ in both their semantics and their typical recommendation objectives. Implicit interactions indicate that a user engaged with an item but provide limited information about the strength or outcome of that preference,

whereas explicit ratings directly encode evaluative information. Earlier unified recommendation approaches showed that combining these feedback types could improve recommendation performance (Liu et al., 2010; Jadidinejad et al., 2019), but they commonly established connections through shared representations, transformations between feedback types, or joint optimisation objectives. Some studies further incorporated neighbour information across feedback types (Bell and Koren, 2007; Koren, 2008a). The question addressed by IEL was therefore more specific: whether users and items could retain feedback-specific representations while allowing these representations to exchange information through their collaborative neighbourhoods. This motivated the research question: **How can implicit and explicit feedback be integrated to learn user and item representations that preserve feedback-specific information and support both item ranking and rating prediction?**

To address this question, IEL represented each user and item separately under implicit and explicit feedback and constructed two corresponding user-item graphs. The distinction between the two main methodological contributions of IEL is important. Conceptually, IEL treated the four feedback-specific representations as *distinct yet interdependent*, rather than forcing the same user or item embedding to represent both feedback types. Technically, GCN-based inter-graph message passing was used to implement this interdependence through neighbour information, whereas intra-graph message passing captured collaborative relationships within each feedback type. The contribution was therefore not simply the use of multiple embeddings or the use of GCNs, but the use of graph propagation to connect feedback-specific representation spaces while preserving their differences.

The selection of implicit item embeddings and explicit user embeddings as the main bridges for inter-graph information exchange reflected a modelling hypothesis about how behavioural and evaluative signals could complement each other. For an implicit item, its explicit user neighbours represented users who had provided direct evaluations of that item and therefore supplied information that was absent from the binary interaction alone. Conversely, the implicit item neighbours of an explicit user represented the user's broader interaction history and could provide behavioural context for the relatively sparse explicit ratings. Inter-graph aggregation was performed before intra-graph propagation so that cross-feedback information could subsequently be propagated through the collaborative neighbourhoods of the corresponding graph. This design provided a relatively simple mechanism for allowing each representation

to access information originating from the other feedback type. Nevertheless, the selected bridge nodes and propagation order were design choices rather than uniquely established solutions, and alternative cross-graph connections were not exhaustively evaluated in this study.

The model development process also provided insight into the appropriate complexity of graph propagation in this setting. The initial designs incorporated feature transformations, non-linear activations, attention-based neighbour aggregation, and edge learning, but these components did not improve ranking performance. The final architecture therefore used simpler propagation and interaction functions, following the general design principle of He et al. (2020). This result suggested that the benefit of IEL arose primarily from how collaborative information was structured and exchanged across feedback-specific graphs, rather than from increasingly expressive transformations of the node embeddings. In other words, introducing cross-feedback information did not necessarily require a more complex GCN architecture to be useful.

The experimental results further showed that modelling feedback-specific embeddings as interdependent was beneficial relative to the compared single-feedback and unified baselines. Within the evaluated configurations, incorporating higher-order neighbours also improved representation learning in several settings. This finding indicated that the relationship between implicit and explicit feedback was not restricted to direct correspondence between the same user or item. Information propagated through neighbouring users and items could also contribute to the representation of a target entity. The significance of this result is that feedback integration can be performed at the representation and neighbourhood levels, rather than only through shared parameters or the combination of ranking and rating losses.

The benefits of the two feedback types were nevertheless asymmetric. Explicit feedback contributed substantially to ranking performance. A likely explanation is that rating values retained preference intensity that was absent from binary interactions, allowing the model to distinguish between items with different levels of user satisfaction rather than treating all observed interactions equally. This interpretation was also reflected in the ranking results, where incorporating rating information improved the ordering of items according to users' expressed preferences.

In contrast, implicit feedback produced considerably smaller improvements in rating prediction. This result should be interpreted in light of the data construction used in this study. The implicit graph was generated by binarising the same observed ratings used to construct the explicit graph. Consequently, the two graphs contained the

same user-item edges, while the implicit representation discarded the rating magnitude. The implicit graph therefore provided a different representation of the observed interactions, but did not introduce additional behavioural observations such as views, clicks, add-to-cart actions, or purchases. Its limited contribution to rating prediction does not imply that independently collected implicit behaviour would generally be uninformative for explicit preference prediction. Instead, the result showed that a binary transformation of the same rating observations provided relatively little additional information for estimating rating values. The RnR results (Hadaş et al., 2018), where the strongest rating prediction performance on both datasets was obtained with a ranking-objective weight of 0.1, showed a similar asymmetry.

These findings also place IEL within the broader development of multi-behaviour recommendation. Recent work increasingly preserves behaviour-specific interaction structures while learning dependencies across different behaviours, rather than collapsing heterogeneous behaviours into a single undifferentiated signal (Zhao et al., 2024a; Kim et al., 2026). IEL shared this general motivation but addressed a different setting: it explicitly connected implicit and explicit feedback representations and jointly considered two recommendation objectives, item ranking and rating prediction. Many multi-behaviour approaches instead use auxiliary behaviours such as clicks or add-to-cart actions to improve the prediction of a designated target behaviour. The findings of IEL therefore supported a broader principle that heterogeneous feedback should retain its feedback-specific semantics while still permitting information transfer between representation spaces.

3.5.1 Limitations and Future Work

The experimental findings also revealed several limitations that define clear directions for future work. The first limitation concerned the construction of implicit feedback. Following previous recommendation studies, the implicit graph was obtained by converting observed ratings into binary interactions. The implicit and explicit graphs therefore had identical connectivity, and the experiments did not reproduce the larger volume or behavioural diversity of implicit feedback commonly observed in real systems. In addition, potentially different implicit behaviours were represented by a single binary relation. This simplification prevented IEL from distinguishing, for example, a view from an add-to-cart action or a purchase, even though these behaviours can express different levels and forms of user intent.

A direct extension would therefore evaluate IEL using independently recorded multi-behaviour data and maintain different behaviour types as separate relations or behaviour-specific graphs. Research on micro-behaviour modelling has demonstrated the value of retaining fine-grained information about user actions, while multi-behaviour recommendation has shown the importance of preserving behaviour-specific signals and modelling dependencies across behaviour types (Zhou et al., 2018b; Yuan et al., 2022; Zhao et al., 2024a; Kim et al., 2026). These studies provide useful starting points for extending IEL from two feedback representations to multiple behaviour-specific representations. Such an extension would allow the model to examine not only whether implicit and explicit feedback are complementary, but also which behavioural signals should transfer information to one another and how their contributions vary across recommendation objectives.

A second limitation was that the same basic GCN propagation mechanism was applied to the implicit and explicit graphs, despite their different information content. This design treated the connectivity structure as the primary signal and did not fully exploit the ordinal information carried by rating values during graph propagation. The stronger ranking results and weaker rating prediction results suggested that this symmetric treatment might have favoured the implicit-feedback task. Recent work on explicit-feedback GCNs has similarly argued that rating graphs require mechanisms that distinguish different rating relations, and has explored rating-specific sub-graphs and rating-aware embeddings for high-order propagation (Peng et al., 2025). Future work could therefore replace the current explicit-graph propagation with rating-aware message passing in which messages, edge representations, or relation-specific parameters depend on the observed rating value. This would preserve the simplicity of the implicit graph while allowing the explicit graph to reflect the richer semantics of numerical ratings.

A third limitation concerned the manually specified inter-graph architecture. IEL selected particular representations as bridges and fixed the order in which inter-graph and intra-graph propagation occurred. Although this design produced favourable results, the experiments did not establish that these connections or this propagation order were optimal. Recent multi-behaviour graph models have increasingly learned behaviour dependencies adaptively rather than relying entirely on fixed transfer structures (Zhao et al., 2024a). Future work could systematically evaluate alternative cross-graph pathways and directions, or introduce learnable gating or attention mechanisms that determine how much information should be transferred between feedback-specific

representations. Such experiments would provide stronger evidence about whether the effectiveness of IEL resulted from the particular information flow used in this study or from the more general principle of cross-feedback neighbour aggregation.

Finally, IEL modelled whether interactions and ratings differed, but it did not explain *why* a behavioural signal and a subsequent explicit evaluation might diverge. For example, a user could purchase an item, indicating initial interest, but later assign a low rating because a particular product characteristic did not meet expectations. Treating the low rating simply as a stronger or weaker numerical signal would not reveal this underlying reason. A natural extension would combine the cross-feedback structure of IEL with richer explicit information such as textual reviews and aspect-sentiment signals. Review-based recommendation research has demonstrated that fine-grained aspect information can expose user preferences and item characteristics that are hidden by overall ratings (Zhang et al., 2014a). Connecting such information with behaviour-specific representations could allow future models to identify not only that implicit and explicit feedback differ, but also which item aspects contributed to the discrepancy. This direction also provides a natural connection between IEL and the aspect-level representation learning investigated in Chapter 4.

The present experiments were additionally restricted to two Amazon product categories. Although the datasets differed in size and density, they did not establish that the observed cross-feedback relationships would generalise to substantially different recommendation domains. Future evaluation should therefore include datasets containing genuine multi-behaviour logs and explicit preference signals across different domains. This would allow the proposed extensions to be evaluated under more realistic differences in feedback quantity, semantics, sparsity, and temporal patterns.

3.6 Conclusions

This study proposed IEL, a GCN-based unified recommendation model that represented users and items using distinct but interdependent embeddings under implicit and explicit feedback. Rather than forcing the two feedback types to share a single representation, IEL preserved their different semantics and connected the corresponding representation spaces through intra-graph and inter-graph message passing. The experimental results showed that this design outperformed the compared baselines in item ranking and also improved rating prediction, although the gains were substantially stronger for ranking.

The main finding was that implicit and explicit feedback were complementary but not equally informative for different recommendation objectives. Explicit ratings provided useful preference-strength information for ranking, whereas the binary feedback derived from the same rating observations provided limited additional information for predicting rating values. The results therefore suggested that unified recommendation models should preserve feedback-specific information and should not assume that information transfer between feedback types is symmetric.

IEL answered the research question by demonstrating that distinct feedback-specific representations could be learned jointly and connected through collaborative neighbourhood information to support both ranking and rating prediction. More broadly, the study showed that integrating heterogeneous feedback can involve relationships between representation spaces and their neighbourhoods, rather than relying only on shared embeddings or joint optimisation objectives. Future work should validate this principle using independently collected multi-behaviour feedback and develop propagation mechanisms that more explicitly reflect the semantics of different feedback types.

Chapter 4

DRL: Dynamic Representation Learning

User-generated textual reviews provide fine-grained information about users' opinions towards item characteristics and have been widely used to complement user-item interaction data in recommender systems (Zhang et al., 2014a; Wang et al., 2021). Review-based recommendation methods can operate at different levels of granularity. Topic-level methods extract latent themes from reviews (Hou et al., 2019; Wang et al., 2023), whereas review-level methods encode entire reviews or review sequences (Luo et al., 2019; Sun et al., 2021; Zhang et al., 2023a). Aspect-level methods instead focus on specific item attributes and, where applicable, the sentiments expressed towards them (Zhang et al., 2014a; Bauman et al., 2017; Wang et al., 2018; Chen et al., 2020b; Zhao et al., 2023). In aspect-based sentiment analysis (ABSA), an aspect may be explicitly mentioned in the text or implicitly inferred from its context (Hua et al., 2024). Aspect-level recommendation can therefore provide structured information about which item characteristics users discuss and how they evaluate those characteristics. Earlier recommendation studies commonly extracted such information using sentiment analysis tools such as Sentires¹ (Zhang et al., 2014b).

Most aspect-level recommendation models derive aspect information from reviews through two main steps. The first step involves extracting structured data related to aspects from the text. This includes processes such as noise filtering, aspect identification, and sentiment analysis. Figure 4.1 illustrates an example review where the user assigned a perfect 5-star rating and provided textual comments to explain their evaluation. The first sentence, "I saw the ad and I was in need of new PS2 memory cards, so I ordered them" offers a broad context but does not convey any specific evaluation of

¹<https://github.com/evison/Sentires>



Reviewed on 17 September 2012

I saw the ad and I was in need of new PS2 memory cards, so I ordered them. The company shipped them out promptly. They arrived in my mailbox in excellent condition. Fantastic job!!!



(delivery, prompt, +1), (condition, excellent, +1)

FIGURE 4.1: An example review from Amazon.com, along with two extracted triples (aspect, opinion phrase, sentiment polarity). Here, sentiment polarity is assigned a value of +1 for positive sentiment and -1 for negative sentiment.

the item. The latter part of the review, however, details the reasons behind the user's satisfaction, from which aspects of the product, corresponding opinion phrases, and inferred sentiment polarities can be extracted. For instance, in this review, the user explicitly highlights two aspects, delivery and condition, indicating that the product performed well in these areas. The second step involves leveraging this structured data to construct user and item representations at the aspect level through various methods (Wang et al., 2018; Xiong et al., 2021). In EFM-style formulations (Zhang et al., 2014a), for example, a user's attention to an aspect is estimated from how frequently the user mentions it, while an item's quality on that aspect is derived from the frequency and sentiment of its mentions. Such structured aspect-level information provides an interpretable representation of user preferences and item characteristics that can subsequently be incorporated into recommendation models.

Aspect-level recommendation models can also support recommendation explanations by exposing specific item characteristics that are associated with the predicted preference (Zhang et al., 2014a). The representation used by a model often influences the form of explanation it can provide (Zhang and Chen, 2020). Topic-level methods may present representative words or topics (Hou et al., 2019), whereas review-level methods can generate or select natural-language content associated with the recommendation (Sun et al., 2020). Aspect-level explanations instead present a small set of identifiable item attributes, allowing the explanation to be directly related to the user's modelled preferences and the item's characteristics. This structured and concise form of explanation motivated the focus on aspects in this study.

In this study, **aspect-level explainable recommendation** refers to jointly predicting a user's overall rating and the positive aspects that the user is likely to associate with a target item. Jointly modelling these outputs is non-trivial because the learned representations need to capture both overall user-item preference and fine-grained aspect-level information. Previous studies have addressed this problem in different ways. Several methods (Zhang et al., 2014a; Gao et al., 2019; Luo et al., 2019; Chen et al., 2020b) combined ID-based latent representations with information derived from reviews, with the review-related component supporting aspect prediction. In contrast, Wang et al. (2018) incorporated the overall rating as a dummy aspect and represented rating and aspect information within a unified prediction framework.

Despite these advances, an important question remains regarding how user and item representations should depend on the current interaction. Several aspect-level models construct general user and item representations from their historical reviews and interactions, and use these representations across different target pairs. AARM (Attentive Aspect-based Recommendation Model) (Guan et al., 2019) provides an important exception by modelling target-aware interactions between user and item aspects through attention. More broadly, target-aware recommendation research has shown that the information used to represent a user can vary according to the candidate item (He et al., 2018b; Cao et al., 2023; Kim et al., 2024). For example, different historical preferences may become relevant when the same user evaluates different target items.

This study investigated a related but more specific problem in aspect-level recommendation: whether both the user and the item could be represented according to their current interacting counterpart by combining collaborative information with fine-grained aspect-sentiment information. For example, a user may place greater emphasis on "graphics" when considering one video game but on "story" when considering another, while different users may attend to different characteristics of the same game. Throughout this chapter, we use the term **dynamic** representation specifically to denote such *interaction-dependent* representation, rather than temporal changes in representations over time. This motivated the following research question: **How can user and item representations be conditioned on each other to support both rating prediction and aspect-level recommendation explanation?**

To address this research question, we developed DRL (Dynamic Representation Learning) with three main components. First, a rating-based GNN aggregated historical rating information from neighbouring users and items to learn general user and

item embeddings. Second, an edge encoder represented the aspect-sentiment information extracted from each review in the same latent space. The dynamic representation layer then used attention to combine the general and edge representations associated with the interacting counterpart, producing an item-dependent user representation and a user-dependent item representation. Finally, the resulting representations were used to predict the target rating and the positive aspects that the user was likely to mention for the item.

For aspect-sentiment extraction, we used a pre-trained LLM, specifically ChatGPT, through a prompt-based procedure to transform textual reviews into structured aspect-sentiment information. Recent research has investigated LLMs for compound ABSA tasks, including the extraction of aspects, opinions, and sentiment polarities, although current evidence does not indicate that LLMs are universally superior to specialised supervised ABSA models (Bai et al., 2024; Hua et al., 2024). As detailed later in this chapter, the LLM-based procedure was adopted after a preliminary Sentires analysis revealed several practical limitations for the Video Games corpus. This preliminary analysis motivated the methodological choice but did not constitute a controlled comparison between the two extraction approaches.

Using the resulting aspect-sentiment data, we evaluated DRL on three public Amazon review datasets. Models requiring aspect information were supplied with aspects derived from the same LLM-based extraction pipeline, so differences in their recommendation performance were not attributable to different aspect extraction methods. The experiments compared DRL with existing recommendation and aspect-level baselines and included ablation studies of its main architectural components. DRL achieved lower MSE/MAE than the compared baselines across all three datasets and higher positive-aspect prediction performance under the offline Hit@N and Recall@N metrics. These results provided empirical support for interaction-dependent representation learning under the evaluation setting used in this study.

Our main contributions can be summarised as:

- We developed DRL, an aspect-level recommendation framework that combined general collaborative representations with interaction-specific aspect-sentiment information to construct interaction-dependent user and item representations for rating and aspect prediction.
- Across three public Amazon datasets, DRL reduced MSE by an average of 5.1%

relative to the strongest compared baseline and improved Hit@1 for positive-aspect prediction by an average of 4.8%. The ablation experiments further examined the contributions and limitations of the main architectural components.

- We used an LLM-based extraction procedure to construct structured aspect-sentiment data from three public review datasets. These datasets supported the aspect-level recommendation and evaluation conducted in this study and provide resources for further research.

4.1 Related Work

Chapter 2 provided a broader review of recommendation methods, review-based recommendation, and explainable recommendation. The purpose of this section is narrower: it focuses on studies most directly related to the design of DRL. Specifically, we examine aspect-level representation learning, target-aware and interaction-dependent representations, and methods for generating and evaluating recommendation explanations. This focused discussion positions DRL relative to the methodological choices most relevant to this chapter rather than repeating the broader literature reviewed in Chapter 2.

In recent years, aspect-based recommendation models (see Table 4.1) have gained significant attention for their ability to capture fine-grained user preferences and item characteristics, enabling more personalised and meaningful recommendations. Some of these models not only predict user-item interactions but also enhance transparency and user trust by providing explanations that highlight specific aspects contributing to a recommendation. The development of these models encompasses a variety of methodologies for generating user-item ratings and aspect-level explanations, reflecting the increasing emphasis on explainable recommendation systems.

4.1.1 Aspect-Level Models

To the best of our knowledge, Zhang et al. (2014a) pioneered aspect-level explainable recommendation. Inspired by matrix factorisation techniques applied to the user-item rating matrix, Zhang et al. (2014a) developed a user-aspect attention matrix and an item-aspect quality matrix to jointly learn explicit user factors, explicit item factors, and aspect factors. Acknowledging the limitations of explicit factors in fully explaining a rating, they additionally incorporated latent user and item factors, which were

TABLE 4.1: Representative models utilising reviews.

Model	Review Granularity	GNN	Rec Obj	Explanations
EFM (Zhang et al., 2014a)	Aspect-level (Aspect, Sentiment)	-	Both	For users (Aspect-based template)
MTER (Wang et al., 2018)	Aspect-level (Aspect, Opinion, Sentiment)	-	Ranking	For users (Aspect-Opinion-based template)
DEAML (Gao et al., 2019)	Aspect-level (Aspect, Sentiment)	-	Rating	For users (Aspect-based template)
HDE (Luo et al., 2019)	Review-level (Aspects used as supervision for review embeddings)	-	Rating	For model designers (Aspect hit)
AARM (Guan et al., 2019)	Aspect-level (Aspect)	-	Ranking	-
A2CF (Chen et al., 2020b)	Aspect-level (Aspect, Sentiment)	-	Ranking	For users (Aspect-based template)
MA-GNNs (Zhang et al., 2023a)	Review-level (Aspects used as indicators for learning review embeddings)	GAT	Ranking	-
ContGraph (Zhao et al., 2023)	Aspect-level (Aspect)	Bayesian GAT	Both	-
DRL (our method)	Aspect-level (Aspect, Sentiment)	GNN	Rating	For users (Aspect-based template)

concatenated with their respective explicit factors to better model the rating. In Zhang et al.’s work, user attention was quantified based on the frequency with which aspects were mentioned in user reviews, while item quality was captured through the frequency of aspects appearing in item reviews and the average sentiment associated with those aspects. Building on this intuitive and effective foundation, numerous subsequent studies (Wang et al., 2018; Gao et al., 2019; Chen et al., 2020b) adopted their approach to develop user and item representations at the aspect level. Our proposed model also leveraged this methodology, which will be elaborated upon in Section 4.2.

Wang et al. (2018) extended Zhang et al. (2014a) by introducing the dimension of the opinion phrase. Specifically, they constructed three tensors: a user-item-aspect tensor for the final recommendation task, and two additional tensors, user-aspect-opinion and item-aspect-opinion. Notably, they incorporated the rating as a dummy aspect appended to the user-item-aspect tensor. Inspired by this approach, our model also adopted a similar method and further investigated its effectiveness. Gao et al. (2019) integrated a hierarchical aspect structure to capture the semantic hierarchical relationships among aspects. For instance, the child aspect "beef" is represented as a sub-concept of the parent aspect "meat". Given the limited expressiveness of matrix

factorisation-based methods, which model pairwise interactions in a linear manner, Chen et al. (2020b) employed a residual feed-forward network to estimate the unobserved values in the user-aspect attention matrix and the item-aspect quality matrix. While Luo et al. (2019) represented users and items at the review level, utilising review embeddings as input, they employed the user-aspect attention vector and item-aspect quality vector as supervisory signals for optimising the review embeddings. Consequently, we regard their approach as falling under aspect-level methods.

Guan et al. (2019) incorporated semantic aspect embeddings and modelled interactions between the aspects associated with a user and those associated with a target item. Its attention mechanism allowed the relative importance of user-side aspects to depend on the candidate item, making it particularly relevant to the interaction-dependent problem investigated in this chapter. However, Guan et al. (2019) used aspect information without explicitly incorporating the corresponding sentiment polarities and was originally designed for top-N recommendation. DRL differed by combining general collaborative representations with aspect-sentiment information and by constructing interaction-dependent representations for both users and items for rating and aspect prediction.

As Graph Neural Networks (GNNs) have proven effective in learning embeddings by aggregating high-order neighbour information, several methods have integrated this technology into aspect-level recommendation models. Zhang et al. (2023a) instead used aspects as indicators for learning review representations and constructed aspect-aware user and item representations through GAT-based propagation. Zhao et al. (2023) similarly incorporated users, items, and aspects into graph-based representation learning. These methods demonstrated the value of graph propagation for incorporating aspect information, but their final recommendation representations were not explicitly constructed according to the current user-item counterpart. In DRL, the GNN was therefore used to learn general collaborative representations, which were subsequently conditioned on the target interaction through the dynamic representation layer.

4.1.2 Target-Aware and Interaction-Dependent Representations

The idea that a user's representation should depend on the current recommendation target predates this study. NAIS (He et al., 2018b) used attention to assign different importance to items in the user's interaction history according to the target item. RTANet

(Cao et al., 2023) extended this principle to network representation learning by constructing user embeddings that varied according to the recommendation target. More recently, MONET (Kim et al., 2024) introduced target-aware attention into multimedia recommendation, using the multi-modal features of a candidate item to identify historical information that was particularly relevant to that target. These studies provided evidence that a single target-independent user representation may not capture all preference information relevant to individual recommendation decisions.

DRL was related to this line of research but addressed a different setting. Rather than conditioning only the user representation on a target item, DRL constructed both an item-dependent user representation and a user-dependent item representation. It also incorporated review-derived aspect-sentiment information into this conditioning process and used the resulting representations for both rating prediction and aspect-level explanation.

4.1.3 Explainable Recommendation Models

Aspect-level models differ in their objectives for providing recommendation explanations, the methods they employ, and the approaches used to evaluate the quality of these explanations. Zhang et al. (2014a) and Gao et al. (2019) utilised explanations to persuade users to click on (or purchase, etc.) the recommended item, focusing on the objective of persuasiveness. Specifically, for each user and a recommended item, Zhang et al. (2014a) chose the aspect in which the item performed best among the user's most cared k aspects, using it as a personalised explanation. To evaluate the impact of these explanations on user behaviour (e.g., Click Through Rate), they conducted an online experiment on an e-commerce website. Gao et al. (2019) introduced a utility-based method to select aspects with the highest utility scores as the recommendation explanation. To assess the persuasiveness of their explanations, they conducted a user study to compare their method against other baselines. Nevertheless, both online experiments and user studies often face constraints such as labour intensity, time consumption, and sampling biases, potentially limiting their feasibility for large-scale applications (Pan et al., 2020).

Moreover, some research has adopted quantitative approaches to measure the quality of generated explanations. Luo et al. (2019) designed a metric to determine the percentage of recommended items for which the model can successfully generate an

explanation. Under their approach, for each user and recommended item, if there exists at least one aspect in the user's most cared aspects where the recommended item outperforms the not-recommended items, the recommendation is deemed explainable. Notably, their method does not rely on the ground truth, namely the aspects of the review in the test set. However, if aspect-level models can predict the aspects that matter to users and where the item excels, information retrieval metrics (e.g., Precision@ k , Recall@ k , etc.) can be applied to check whether these aspects do indeed appear in the ground truth (Tan et al., 2021). Wang et al. (2018) evaluated their model by selecting the top 20 aspects and top 50 opinion phrases as recommendation explanations. Yet, choosing too large a k can compromise conciseness and usability, given users' limited time and attention for reading lengthy explanations (Chen et al., 2020b). In this study, we therefore selected only a small number of aspects (fewer than five) as our recommendation explanation. Details of the evaluation procedure are provided in Section 4.4.2.

The present study therefore focused on a narrower explanation form. Rather than generating unrestricted natural-language explanations, DRL predicted a small number of aspects linked directly to its modelled user preferences and item characteristics. This provided a structured and model-grounded explanation target that could be evaluated offline against aspects appearing in held-out reviews. As discussed later in this chapter, this evaluation measures aspect-prediction performance and should not be interpreted as a direct measure of user-perceived explanation quality.

4.2 Preliminaries

This section introduces the method proposed in EFM (Zhang et al., 2014a) for representing users and items at the aspect level, based on aspects and their corresponding sentiments extracted from reviews. Many subsequent studies adopt this approach by constructing two matrices to represent each user's attention to aspects (the User-Aspect Attention Matrix, see Section 4.2.1) and each item's quality across aspects (Item-Aspect Quality Matrix, see Section 4.2.2). In this study, we also adopted this intuitive and effective method. Specifically, aspect-sentiment pairs (a, s) were derived from each textual review using aspect-based sentiment analysis. For positive sentiments, s was assigned a value of +1, while for negative sentiments, s was assigned a value of -1. For instance, the movie review "The special effects were what I expected, but the plot is boring" can be processed into two aspect-sentiment pairs: (*special effects*, +1) and (*plot*, -1). After

processing all reviews, an aspect set $\mathcal{A} = \{a_1, a_2, \dots, a_p\}$ was generated, encompassing all aspects mentioned within the dataset, where p is the total number of aspects. Finally, for each user, we counted the total number of times they mentioned each aspect. Meanwhile, for each item, we summed the sentiment polarities it received for each aspect.

4.2.1 User-Aspect Attention Matrix

Each element in the user-aspect attention matrix quantifies the degree to which a user is concerned with a specific aspect:

$$X_{ih} = \begin{cases} 0, & \text{if user } u_i \text{ does not mention aspect } a_h, \\ 1 + (N - 1)(\frac{2}{1+e^{-t_{ih}}} - 1), & \text{otherwise,} \end{cases} \quad (4.1)$$

where t_{ih} represents the frequency with which user u_i has mentioned aspect a_h across all their reviews, and $[1, N]$ denotes the range of ratings that users can assign on the platform. In many real-world five-star reviewing systems, such as Amazon and Yelp, N is typically set to 5.

4.2.2 Item-Aspect Quality Matrix

Each element in the item-aspect quality matrix quantifies the quality of an item with respect to a specific aspect:

$$Y_{jh} = \begin{cases} 0, & \text{if aspect } a_h \text{ of item } v_j \text{ is not mentioned,} \\ 1 + \frac{N-1}{1+e^{-s_{jh}}}, & \text{otherwise,} \end{cases} \quad (4.2)$$

where s_{jh} represents the aggregated sentiment polarity for aspect a_h across all reviews of item v_j . Similar to Equation 4.1, the value is also rescaled to fall within the range $[1, N]$.

4.3 Proposed Model

In this study, we propose a Dynamic Representation Learning (DRL) model for users and items. The architecture of DRL is depicted in Figure 4.2. Our model consists of

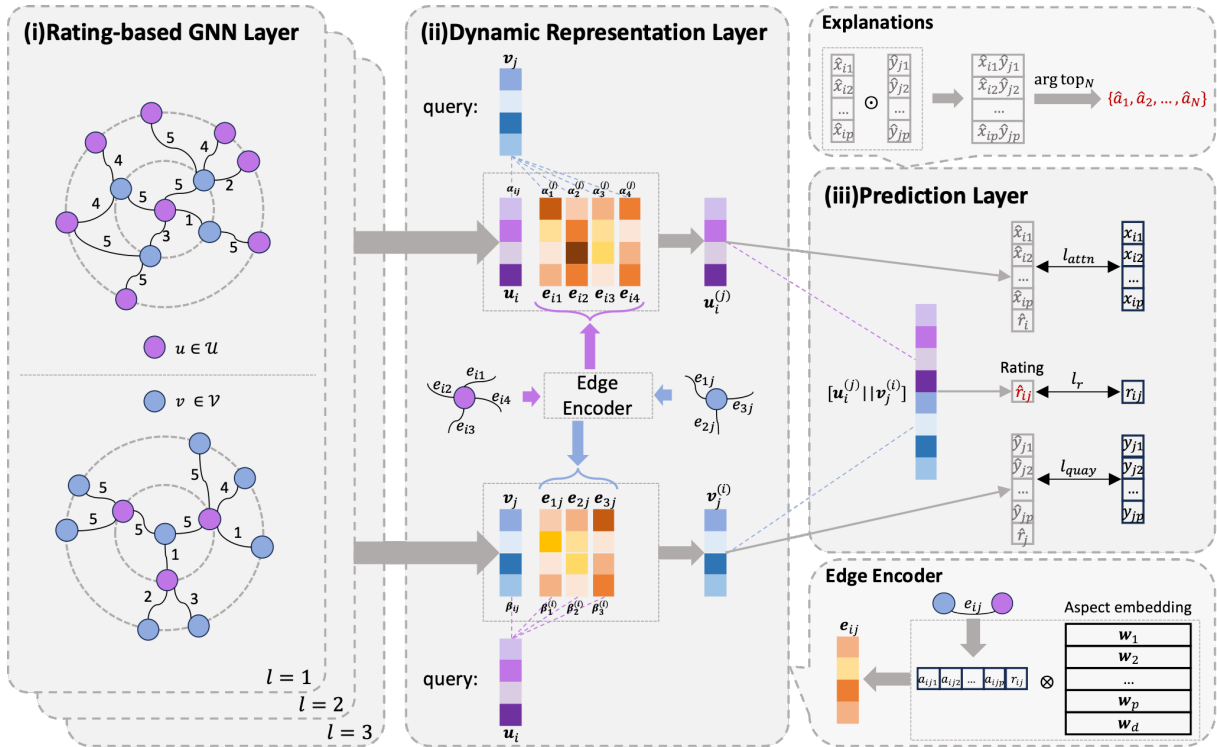


FIGURE 4.2: The architecture of DRL. (i) A rating-based GNN layer that generates general embeddings for users and items. (ii) A dynamic representation layer that utilises an edge encoder to transform aspect information into edge embeddings and an attention mechanism to generate interaction-dependent embeddings. (iii) A prediction layer that integrates the user and item interaction-dependent embeddings to predict the rating and identify the positive aspects.

three layers: (i) a rating-based GNN layer, which aggregates high-order neighbour information for users/items based on historical ratings to update their representations; (ii) a dynamic representation layer, designed to capture the user embedding in the context of the specific item and the item embedding in the context of the particular user; (iii) a prediction layer, which includes the rating prediction for the target user and item, as well as predictions for their respective aspect attention vector and aspect quality vector. A key design choice is that aspect information is not introduced as a third node type in the collaborative graph. Instead, the rating-based GNN is used to learn general collaborative representations from user-item interactions, while review-derived aspect-sentiment information is retained at the interaction level and incorporated by the subsequent dynamic representation layer. This separation allows the model to examine interaction-dependent representation learning without simultaneously introducing a user-item-aspect graph structure. The details of these three components will be elaborated in the following sections.

4.3.1 Rating-Based GNN Layer

We argue that different neighbouring nodes contribute to the target node with varying degrees of importance. Thus, when aggregating information from neighbours, we assigned different weights to each neighbouring node. An inherent source of user-labelled information, such as rating values, can serve as the weight for neighbouring nodes. For example, within a user's rating history, items rated with 5 stars should better reflect the user's preferences than those rated with 1 star. Similarly, among the users who rate an item, those who give higher ratings may provide more representative information about the item's characteristics than those who give lower ratings. Therefore, when aggregating a user's neighbour information at each GNN layer, we assigned different weights to each item based on its softmax-transformed historical rating. The same approach was applied for each item's aggregation of user information.

To further elaborate, we now introduce the rating-based GNN from two perspectives: the user-side GNN and the item-side GNN, which separately generate user embeddings and item embeddings across different layers. Once the user-side and item-side embeddings at each layer have been computed, we independently integrated the representations on each side by aggregating all layers, thereby deriving final embeddings for both users and items. We treated the user and item embeddings generated

from the GNN layer as the inherent features of users and items. Therefore, we refer to them as the user/item general embedding.

User-Side GNN

$$\begin{aligned} \mathbf{u}_i^{(l+1)} &= \sum_{k \in \mathcal{N}_i} r'_{ik} \cdot \mathbf{v}_k^{(l)}, \\ r'_{ik} &= \frac{\exp(r_{ik})}{\sum_{k' \in \mathcal{N}_i} \exp(r_{ik'})}, \end{aligned} \quad (4.3)$$

where $\mathbf{u}_i^{(l+1)}$ denotes u_i 's embedding at layer $l + 1$, $\mathcal{N}_i$ is the user's interaction history set, and r_{ik} represents the rating u_i gives to v_k .

Item-Side GNN

$$\begin{aligned} \mathbf{v}_j^{(l+1)} &= \sum_{t \in \mathcal{N}_j} r'_{tj} \cdot \mathbf{u}_t^{(l)}, \\ r'_{tj} &= \frac{\exp(r_{tj})}{\sum_{t' \in \mathcal{N}_j} \exp(r_{t'j})}, \end{aligned} \quad (4.4)$$

where $\mathbf{v}_j^{(l+1)}$ denotes v_j 's embedding at layer $l + 1$, $\mathcal{N}_j$ is the item's interaction history set, and r_{tj} represents the rating that u_t gives to v_j .

Layer Combination

After obtaining the embeddings from each layer, we used a self-attention mechanism to integrate the information from each layer to produce the final embedding:

$$\mathbf{u}_i = \sum_{l=0}^L \alpha_{i,l} \mathbf{u}_i^{(l)}, \quad (4.5)$$

$$\mathbf{v}_j = \sum_{l=0}^L \beta_{j,l} \mathbf{v}_j^{(l)}, \quad (4.6)$$

with $\alpha_{i,l}$ and $\beta_{j,l}$ are computed as follows:

$$\alpha_{i,l} = \frac{\exp(\alpha'_{i,l})}{\sum_{l=0}^L \exp(\alpha'_{i,l})}, \quad \alpha'_{i,l} = \boldsymbol{\alpha}^\top \text{LeakyReLU}(\mathbf{W}_\alpha \mathbf{u}_i^{(l)} + \mathbf{b}_\alpha), \quad (4.7)$$

$$\beta_{j,l} = \frac{\exp(\beta'_{j,l})}{\sum_{l=0}^L \exp(\beta'_{j,l})}, \quad \beta'_{j,l} = \boldsymbol{\beta}^\top \text{LeakyReLU}(\mathbf{W}_\beta \mathbf{v}_j^{(l)} + \mathbf{b}_\beta), \quad (4.8)$$

where $W_\alpha, W_\beta \in \mathbb{R}^{d \times d}$, and $b_\alpha, b_\beta, \alpha_u, \beta \in \mathbb{R}^d$ are learned parameters.

4.3.2 Dynamic Representation Layer

Each user's or item's associated reviews may contribute differently when the entity is considered with different interaction partners. Accordingly, the Dynamic Representation Layer constructs interaction-dependent representations rather than temporally evolving ones. For a target user-item pair, the general embedding of one entity is used as the query to weight the other entity's general embedding and historical review-derived edge embeddings. This allows the user representation to vary across target items and the item representation to vary across target users. The following sections describe the edge encoder and the construction of the interaction-dependent user and item embeddings.

Edge Encoder

Based on the extracted aspects and their corresponding sentiments, each review can be expressed as a list of (a, s) pairs. When encoding each review that u_i gives to v_j , we first represented it as a vector $\mathbf{a}_{ij} = [a_{ij1}, a_{ij2}, \dots, a_{ijp}]$ of length p , where p is the total number of aspects. Each element in the vector is calculated as follows:

$$a_{ijh} = \begin{cases} 0, & \text{if aspect } a_h \text{ is not mentioned by } u_i \text{ about } v_j, \\ 1 + \frac{N-1}{1+e^{-s_{ijh}}}, & \text{otherwise,} \end{cases} \quad (4.9)$$

where s_{ijh} represents the summation of all sentiments that u_i expresses on v_j 's aspect a_h .

This sparse vector can be directly transformed into a dense embedding for subsequent calculations in the Dynamic Representation Layer. However, to better capture semantic information from the reviews, we pre-trained an aspect embedding matrix $W_A \in \mathbb{R}^{d_a \times p}$ and then performed matrix multiplication between this sparse vector and W_A , producing a dense and semantically enriched vector. This approach aims to capture potential similarities and correlations among aspects. Table 4.2 presents the top 10 most frequent aspects across each dataset². Notably, some aspects share semantic resemblance, such as "song" and "music" in Amazon-CDs & Vinyl, as well as singular and plural forms of the same aspect in Amazon-Video Games, like "graphic" and

²The datasets and the process of extracting aspects are described in detail in Sections 4.5.1 and 4.5.2.

"graphics".³ Additionally, there are related aspects, such as "usability" and "functionality" in Amazon-Office. Therefore, considering the semantic information of aspects enables each review to incorporate both sentiment information and an enhanced level of semantic richness. Consistent with previous research (Guan et al., 2019), the parameters of W_A remain fixed during training.

TABLE 4.2: Top 10 of filtered aspects in each dataset.

Dataset	Top 10 Aspects
Office	price, durability, quality, design, color, size, usability, appearance, functionality, performance
Video Games	gameplay, graphic, story, game, price, character, control, difficulty, graphics, multiplayer
CDs & Vinyl	album, song, track, music, cd, lyric, sound, vocal, guitar, band

We argue that, beyond aspect information extracted from reviews, each edge should also encode rating information. Unlike the rating-based GNN layer, where the rating value serves as a weight for neighbour information, we incorporated the rating as an explicit feature within the edge's attributes in the dynamic representation layer. This approach reflects a practical scenario in which users consider both an item's performance across various aspects and its ratings when making decisions. From a modelling perspective, the rating value thus functions not only as a final training target but also as an active feature in model training. Therefore, we treated the rating as a "dummy" aspect when representing edges. Specifically, each edge is initially represented as an aspect vector $\hat{\mathbf{a}}_{ij} = [a_{ij1}, a_{ij2}, \dots, a_{ijp}, r_{ij}]$ of length $p + 1$, with the final element denoting the rating value. Correspondingly, we introduced a dummy embedding $\mathbf{w}_d \in \mathbb{R}^{d_a}$ as a trainable parameter, which multiplies with the dummy aspect (i.e., the rating value) to be integrated into the edge's representation.

$$\hat{\mathbf{e}}_{ij} = \hat{\mathbf{a}}_{ij} \otimes [\mathbf{W}_A || \mathbf{w}_d]^\top, \quad (4.10)$$

where $\otimes$ and $||$ denote matrix multiplication and concatenation operation, respectively. Finally, a trainable matrix $\mathbf{W}_{trans} \in \mathbb{R}^{d \times d_a}$ maps the edge from the pre-trained semantic vector space to the vector space of the recommendation task.

$$\mathbf{e}_{ij} = \mathbf{W}_{trans} \hat{\mathbf{e}}_{ij}. \quad (4.11)$$

³Despite setting extraction rules within the prompt, minor inconsistencies are expected when using a generative language model to extract aspects.

Dynamic User Embedding

We employed an attention mechanism to learn the interaction-dependent user embedding during interactions. Specifically, we used the item's general embedding $\mathbf{v}_j$ learned from the rating-based GNN layer as the query, while the user's general embedding $\mathbf{u}_i$ and the user's edge embeddings $\mathbf{e}_{ik}$ ($k \in \mathcal{N}_i$) learned from the Edge Encoder module serve as the keys and values. In other words, the keys and values for each user remain constant, but different attention scores are generated when interacting with different items (the queries):

$$\hat{\alpha}_{ij} = \boldsymbol{\alpha}^\top \text{LeakyReLU}(\mathbf{W}_v \mathbf{v}_j + \mathbf{W}_\alpha \mathbf{u}_i), \quad (4.12)$$

$$\hat{\alpha}_k^{(j)} = \boldsymbol{\alpha}^\top \text{LeakyReLU}(\mathbf{W}_v \mathbf{v}_j + \mathbf{e}_{ik}), \quad (4.13)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^d$, $\mathbf{W}_v$ and $\mathbf{W}_\alpha \in \mathbb{R}^{d \times d}$ are learned weights. These attention scores are normalised across the target i and all neighbours $k \in \mathcal{N}_i$ using softmax, and the attention function is defined as:

$$\alpha_{ij} = \frac{\exp(\hat{\alpha}_{ij})}{\exp(\hat{\alpha}_{ij}) + \sum_{k' \in \mathcal{N}_i} \exp(\hat{\alpha}_{k'}^{(j)})}, \quad \alpha_k^{(j)} = \frac{\exp(\hat{\alpha}_k^{(j)})}{\exp(\hat{\alpha}_{ij}) + \sum_{k' \in \mathcal{N}_i} \exp(\hat{\alpha}_{k'}^{(j)})}, \quad (4.14)$$

This results in a interaction-dependent user embedding, conditioned on the item's general embedding:

$$\mathbf{u}_i^{(j)} = \text{ReLU}(\alpha_{ij} \mathbf{u}_i + \sum_{k \in \mathcal{N}_i} \alpha_k^{(j)} \mathbf{e}_{ik}). \quad (4.15)$$

Please note that $j \notin \mathcal{N}_i$, indicating we excluded the target edge $\mathbf{e}_{ij}$ from the user's historical edge set.

Dynamic Item Embedding

For learning interaction-dependent item embeddings, we applied the same method as learning interaction-dependent user embeddings but with different parameters. The calculation of attention scores is as follows:

$$\hat{\beta}_{ij} = \boldsymbol{\beta}^\top \text{LeakyReLU}(\mathbf{W}_u \mathbf{u}_i + \mathbf{W}_\beta \mathbf{v}_j), \quad (4.16)$$

$$\hat{\beta}_t^{(i)} = \boldsymbol{\beta}^\top \text{LeakyReLU}(\mathbf{W}_u \mathbf{u}_i + \mathbf{e}_{tj}), \quad (4.17)$$

where $t \in \mathcal{N}_j$, $\beta \in \mathbb{R}^d$, W_u and $W_\beta \in \mathbb{R}^{d \times d}$ are learned weights. Before generating the interaction-dependent embeddings, the attention scores are processed through a softmax function.

$$\beta_{ij} = \frac{\exp(\hat{\beta}_{ij})}{\exp(\hat{\beta}_{ij}) + \sum_{t' \in \mathcal{N}_j} \exp(\hat{\beta}_{t'})}, \quad \beta_t^{(i)} = \frac{\exp(\hat{\beta}_t^{(i)})}{\exp(\hat{\beta}_{ij}) + \sum_{t' \in \mathcal{N}_j} \exp(\hat{\beta}_{t'}^{(i)})}. \quad (4.18)$$

We also excluded the target edge e_{ij} from the item's historical edge set, i.e., $i \notin \mathcal{N}_j$.

$$\mathbf{v}_j^{(i)} = \text{ReLU}(\beta_{ij} \mathbf{v}_j + \sum_{t \in \mathcal{N}_j} \beta_t^{(i)} \mathbf{e}_{tj}). \quad (4.19)$$

4.3.3 Prediction Layer

Our model includes two prediction targets: one is the user's overall rating for the item, and the other is the user's aspect-level review of the item. The aspect-level review represents the user's ratings across various aspects of the item, with higher scores indicating greater satisfaction with the item's performance on that particular aspect. Consequently, we can identify the highest-scoring aspects as reasons for recommending the item to the user. In the following sections, we will provide a detailed description of both the rating prediction and aspect-level review prediction.

Rating Prediction

The interaction-dependent embeddings $\mathbf{u}_i^{(j)}$ and $\mathbf{v}_j^{(i)}$, generated by the Dynamic Representation Layer, capture the interaction-specific information for the user and item, respectively. We concatenated and passed them through two fully connected layers to produce the final predicted rating:

$$\hat{r}_{ij} = \text{ReLU}(\mathbf{W}_2^\top \text{ReLU}(\mathbf{W}_1 \cdot [\mathbf{u}_i^{(j)} || \mathbf{v}_j^{(i)}])), \quad (4.20)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times 2d}$ and $\mathbf{W}_2 \in \mathbb{R}^d$, and $||$ denotes the concatenation operation.

Aspect-Level Review Prediction

For aspect-level review prediction, the interaction-dependent embeddings of the user and item, containing their respective historical review information, are processed through individual fully connected layers to generate the user attention vector $\hat{\mathbf{x}}$ and the item

quality vector $\hat{\mathbf{y}}$. Specifically, $\mathbf{u}_i^{(j)}$ is mapped to a vector $\mathbf{x}'_i$ of length $p + 1$ first:

$$\mathbf{x}'_i = \text{ReLU}(\mathbf{W}_{\text{attn}} \mathbf{u}_i^{(j)}), \quad (4.21)$$

where $\mathbf{W}_{\text{attn}} \in \mathbb{R}^{(p+1) \times d}$ are learned weights. Since aspects and the rating are encoded within the edge information, we treated the first p values of $\mathbf{x}'_i$ as the predicted user attention vector $\hat{\mathbf{x}}_i = \mathbf{x}'_i[: -1] = [\hat{x}_{i1}, \hat{x}_{i2}, \dots, \hat{x}_{ip}]$ for aspects. The last value can be seen as the user's rating tendency, which is discarded in our model. For predicting the item quality vector $\hat{\mathbf{y}}_j = \mathbf{y}'_j[: -1] = [\hat{y}_{j1}, \hat{y}_{j2}, \dots, \hat{y}_{jp}]$, we adopted the similar method:

$$\mathbf{y}'_j = \text{ReLU}(\mathbf{W}_{\text{quay}} \mathbf{v}_j^{(i)}), \quad (4.22)$$

where $\mathbf{W}_{\text{quay}} \in \mathbb{R}^{(p+1) \times d}$ are learned weights.

Finally, we applied a matching approach, specifically the element-wise multiplication of $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{y}}_j$, to generate the user's aspect-level review of the item:

$$\hat{\mathbf{a}}_{ij} = \hat{\mathbf{x}}_i \odot \hat{\mathbf{y}}_j = [\hat{x}_{i1}\hat{y}_{j1}, \hat{x}_{i2}\hat{y}_{j2}, \dots, \hat{x}_{ip}\hat{y}_{jp}]. \quad (4.23)$$

Our recommendation rationale is likewise derived from this predicted aspect-level review. We hypothesised that an aspect should only serve as a recommendation reason if the user values that aspect and the item performs well on it. Accordingly, we ranked aspects based on the product of the user's attention on each aspect and the item's quality for that aspect. The top N aspects with the highest values are then selected to construct the recommendation explanations:

$$\{\hat{a}_{ij1}, \hat{a}_{ij2}, \dots, \hat{a}_{ijN}\} = \arg \text{top}_N(\hat{\mathbf{a}}_{ij}). \quad (4.24)$$

For simplicity, we omitted the user and item subscripts and represented the aspects serving as personalised explanations as $\{\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N\}$.

4.3.4 Model Training

The loss function for training the model consists of three components: rating prediction loss l_r , aspect prediction loss, and a regularisation term l_{reg} . All three use L_2 loss. The aspect prediction loss is composed of the user attention loss l_{attn} and item quality loss

l_{quay} across aspects.

$$l = \frac{1}{|\mathcal{R}_{train}|} \sum_{(i,j) \in \mathcal{R}_{train}} \{l_r + \lambda_1(l_{attn} + l_{quay}) + \lambda_2 l_{reg}\}, \quad (4.25)$$

where $\mathcal{R}_{train}$ denotes the training set consisting of user-item interaction pairs (i, j) , λ_1 and λ_2 are hyperparameters that adjust the contribution of each component to the total loss. The rating prediction loss is calculated as follows:

$$l_r = (\hat{r}_{ij} - r_{ij})^2, \quad (4.26)$$

where $\hat{r}_{ij}$ and r_{ij} are the predicted rating and the real rating, respectively.

For l_{attn} and l_{qual} , we only calculated the non-zero entries in the ground truth values of $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{ip}]$ and $\mathbf{y}_j = [y_{j1}, y_{j2}, \dots, y_{jp}]$:

$$l_{attn} = \sum_{h=1}^p \mathbf{I}(x_{ih} \neq 0) \cdot (\hat{x}_{ih} - x_{ih})^2, \quad (4.27)$$

$$l_{quay} = \sum_{h=1}^p \mathbf{I}(y_{jh} \neq 0) \cdot (\hat{y}_{jh} - y_{jh})^2, \quad (4.28)$$

where $\mathbf{I}(\cdot)$ is an indicator function.

In addition to L_2 regularisation, we also applied dropout following the softmax operation during the generation of interaction-dependent user/item embeddings, as well as after mapping them to the predicted user attention and item quality vectors, to avoid overfitting.

4.4 Explanation Format and Evaluation

As described in Section 4.3.3, our model DRL can generate one or multiple aspects $\{\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N\}$ as personalised recommendation explanations. In this section, we introduce the format of the explanations and the method for evaluating them.

4.4.1 Explanation Format

For items with the predicted rating exceeding 3 ($\hat{r}_{ij} > 3$), we recommend them to users along with relevant recommendation reasons. Building on previous research (Zhang

et al., 2014a) that provides aspect-level explanations, we adopted a similar template to deliver personalised explanations to users when recommending an item. The template is shown as follows:

We recommend this product $[v_j]$ because you may be satisfied with its $[\hat{a}_1]$ and $[\hat{a}_2]$.

We selected the top two aspects for explanation, as our model achieved a hit ratio of the ground truth exceeding 50% across all datasets when providing two aspects.

4.4.2 Explanation Evaluation

To evaluate the explanation performance of our model, we adopted both quantitative methods, based on offline formula-based evaluation, and qualitative analysis through case studies.

For the formula-based evaluation, we used aspects extracted from reviews as the ground truth, a method similar to previous work (Tan et al., 2021). Specifically, from each positive review ($r_{ij} > 3$) that user u_i provided for item v_j , we extracted all aspects with positive sentiments, represented as $\mathbf{p}_{ij} = [p_{ij1}, p_{ij2}, \dots, p_{ijp}]$. Here, $\mathbf{p}_{ij}$ is a binary vector, where each element p_{ijh} equals 1 if user u_i mentioned aspect a_h with a positive sentiment, and $p_{ijh} = 0$ otherwise. For each user-item interaction pair, we employed Hit@N, Recall@N, and Precision@N to evaluate the explainability performance:

$$\text{Hit@N} = \begin{cases} 1, & \text{if } \exists n \in \{1, 2, \dots, N\} \text{ such that } \hat{a}_n \in \{a_h | p_{ijh} = 1\}, \\ 0, & \text{otherwise,} \end{cases} \quad (4.29)$$

$$\text{Recall@N} = \frac{|\{\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N\} \cap \{a_h | p_{ijh} = 1\}|}{\sum_{h=1}^p p_{ijh}}, \quad (4.30)$$

$$\text{Precision@N} = \frac{|\{\hat{a}_1, \hat{a}_2, \dots, \hat{a}_N\} \cap \{a_h | p_{ijh} = 1\}|}{N}. \quad (4.31)$$

In these evaluation metrics, Hit@N denotes whether at least one of the N provided aspect explanations matches a positive aspect expressed in the user's actual review. Recall@N measures the proportion of all positive aspects that are covered by the N selected aspects, while Precision@N represents the proportion of the N selected aspects that are indeed positive. Finally, we reported the average scores of all pairs as the final Hit@N, Recall@N, and Precision@N.

When quantitatively evaluating the quality of recommendation explanations generated by the model, we considered two settings based on different review selection criteria: reviews associated with **actual ratings** greater than 3 versus those associated with **predicted ratings** greater than 3.

First, we used reviews linked to actual ratings greater than 3 as the ground truth. This setting directly reflects the model’s ability to generate explanations for items that users genuinely appreciated. Although in practice the system predicts high-rated items before presenting them with explanations, this evaluation isolates explanation quality by assessing whether the model can accurately capture the positive aspects of items that users truly liked. Importantly, this setting enables fair model comparisons, as it ensures a consistent ground truth by relying on real positive user feedback.

Second, to simulate a real-world deployment scenario in which the model only recommends items with predicted ratings above 3, we additionally evaluated explanation performance based on reviews associated with such items. This supplementary setting assesses whether the explanations are convincing within the scope of items the model itself decides to recommend. However, this approach may introduce inconsistencies in the ground truth when comparing different models. For example, if a model predicts a rating of 3 or below for a user-item pair, that instance would be excluded from the evaluation even when the actual rating was greater than 3 and therefore represented a positive user evaluation. Although prediction errors may introduce noise into this evaluation, it remains valuable for understanding the model’s expected behaviour under real deployment conditions.

In addition to the offline formula-based evaluations, we conducted case studies to provide a more detailed and intuitive understanding of the model’s explanation quality. Specifically, we presented examples showing the predicted ratings and aspects alongside the actual ratings and user reviews. This qualitative analysis complements the quantitative results by illustrating how well the model’s generated explanations align with real user opinions in individual cases, helping to identify strengths and potential limitations that may not be fully captured by aggregate metrics.

4.5 Experiments

In this section, we introduce the experimental setup, including dataset preparation steps such as extracting aspects and their sentiment polarity from reviews, removing

infrequent aspects, and generating semantic embeddings for the remaining valid aspects. Additionally, we outline the key characteristics of the baselines and all parameter configurations. The experimental results encompass a comprehensive evaluation of our model, DRL, against all baselines in terms of rating prediction and explainability, an analysis of the impact of varying GNN layer depths on DRL's performance, and the results of ablation studies conducted to assess the contributions of specific components within our model.

Before adopting the LLM-based extraction procedure, we conducted a preliminary analysis using Sentires, a phrase-level sentiment analysis toolkit that extracts aspect-opinion-sentiment triplets from textual reviews. This analysis was performed on the original, unprocessed Amazon Video Games review corpus and was used to assess the suitability of the extracted aspects for the downstream representation learning task.

Sentires extracted 5,012 distinct aspects from the corpus. However, several issues were observed. First, the extracted vocabulary was highly fragmented. Semantically similar expressions and morphological variants were represented as separate aspects, such as "price", "buy", and "purchase", or "game", "games", "play", "playing", and "plays". Second, some extracted terms were incomplete without their surrounding context. For example, "fan" and "mode" frequently occurred as parts of more informative expressions such as "Zelda fan" or "multiplayer mode". Third, a number of highly frequent extracted terms, such as "play", "product", and "features", provided limited information for distinguishing fine-grained user preferences or item characteristics. Other domain-specific terms were also difficult to interpret consistently as preference-bearing aspects without considering their local context.

The extraction was also sparse at the review level. Of the 497,577 reviews in the Amazon Video Games corpus⁴, 202,157 did not contain an aspect-opinion-sentiment triplet identified by Sentires. Across the extracted vocabulary, users discussed an average of 11.4 aspects and items an average of 24.5 aspects, corresponding to user-aspect and item-aspect densities of 0.23% and 0.49%, respectively. Part of this sparsity may have reflected the nature of game reviews themselves, some of which primarily describe gameplay, context, or personal experiences without expressing an explicit evaluative opinion about a particular item feature. It may also have resulted from limitations of the extraction procedure.

⁴<https://nijianmo.github.io/amazon>

These observations motivated us to explore an LLM-based approach for constructing the aspect-sentiment data used in this study. Rather than relying only on surface-level aspect-opinion expressions, the LLM could be prompted to identify contextually relevant aspects and their sentiment polarities, including aspects that were implied by the review rather than expressed using a predetermined surface form. Recent ABSA research similarly distinguishes explicit aspect terms from implicit aspects inferred from context and has increasingly investigated generative language models for such tasks (Hua et al., 2024). Importantly, the preliminary Sentires analysis was not designed as a controlled benchmark against the LLM-based method. Because it was conducted on the original corpus using a different processing pipeline and without manually annotated ground truth, it did not establish that the LLM extraction was generally more accurate than Sentires. Instead, it provided corpus-specific evidence that motivated the alternative extraction strategy adopted in this thesis.

4.5.1 Extraction of Aspects and Corresponding Sentiment Polarity

```
1 def generate_prompt(dataset, examples, review):
2     return f """
3     Your task is to analyse product reviews and extract product features with their sentiment
4     scores from the {dataset} dataset. You can either extract explicit features mentioned in the
5     review or infer appropriate features from the context.
6     You must strictly adhere to the following rules:
7     1. Normalise features (e.g., convert to lowercase and singular form) and apply appropriate
8     NLP processing such as stemming or lemmatisation.
9     2. Only assign the feature 'overall' if no specific features can be extracted or inferred
10    from the review.
11    3. Follow the output format exactly as shown in the examples: [(feature, sentiment)].
12    4. Do not output your analysis process. Only provide the final features with their sentiment
13    scores.
14    Here are examples:
15    {examples}
16    Please analyse the review below, delimited by triple backticks, and provide the features
17    with their sentiment scores:
18    '''{review}'''
19    """
```

FIGURE 4.3: Python code for generating prompts to extract or infer structured (aspect, sentiment score) pairs from textual reviews. The function takes the current dataset, examples, and the target review as inputs, and encodes detailed task instructions, including four explicit rules for the model to follow.

We employed an LLM with prompt engineering to extract aspects and their corresponding sentiment polarity from reviews, using OpenAI's chat completions API⁵ and

⁵<https://platform.openai.com/docs/api-reference/chat/create>

the "gpt-4o" model⁶. The procedure converted each user review into a set of (aspect, sentiment polarity) pairs, denoted as (a, s) pairs. Specifically, we labelled sentiment polarity s as either a positive score (+1) or a negative score (-1). The Python code for generating prompts is shown in Figure 4.3.

We adopted three datasets⁷ of varying scales from the Amazon e-commerce platform: Office, Video Games, and CDs & Vinyl. To ensure that each user and item has sufficient interactions and aspects to construct a comprehensive profile at the aspect level, we removed users and items with fewer than 10 reviews in all datasets. The statistics⁸ for these datasets are presented in Table 4.3. When generating prompts, as illustrated in the code, we provided not only clear and general instructions but also specific examples for each dataset to better guide the generative LLM in producing the desired outputs. The prompt includes three examples, with the first example featuring a real review from the dataset that has been appropriately modified to adhere to the first rule in the prompt. In other words, the first example varied across datasets:

Amazon Office: "These blinders are DURABLE. Nice cover." -> [(durability, +1), (cover, +1)]

Amazon Video Games: "Bought for future investing/collecting." -> [(collection, +1)]

Amazon CDs & Vinyl: "It has some really good tracks on it that are very well done. Its a CHRISTMAS favorite of mine." -> [(track, +1), (christmas, +1)]

The second and third examples are common across all datasets: the second example demonstrates to the LLM that it can infer appropriate features from the reviews, while the third example indicates how the LLM should proceed when a review is brief and specific features cannot be directly extracted or inferred.

Example 2: "I think there might be other options that are nicer for the same price. It arrived a day earlier than expected." -> [(price, -1), (delivery, +1)]

Example 3: "good" -> [(overall, +1)]

Based on the preliminary analysis above, we adopted an LLM-based extraction approach that could be prompted to identify contextually relevant aspects and their associated sentiment polarities. In particular, the approach allowed aspects to be inferred from the review context rather than requiring every aspect to appear verbatim as an extracted surface term. For example, while the phrase "It arrived a day earlier than

⁶<https://platform.openai.com/docs/models#gpt-4o>

⁷<https://nijianmo.github.io/amazon>

⁸In processing the datasets, we removed duplicate interactions, i.e., if a user reviewed the same item multiple times, only the most recent interaction was retained.

expected" does not explicitly mention an aspect, the relevant one "delivery" can be inferred from this statement. For very brief reviews where aspects cannot be extracted or inferred, such as "good", we assigned a general aspect label, "overall". This approach not only preserves these interactions within the dataset but also aligns with certain reviewing habits, where users may avoid mentioning specific aspects and instead provide an overall impression of the item's performance.

TABLE 4.3: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
Amazon-Office	3,612	2,427	58,289	0.00665
Amazon-Video Games	5,932	3,788	103,624	0.00461
Amazon-CDs & Vinyl	21,445	18,392	527,326	0.00134

The statistics of the original reviews and the extracted aspects for each dataset are summarised in Table 4.4. The number of extracted aspects in the Office, Video Games, and CDs & Vinyl datasets is 10,686, 29,738, and 85,381, respectively. Here, positive and negative reviews are defined as those with corresponding ratings $r_{ij} > 3$ and $r_{ij} \leq 3$, respectively. The column "#Pos Aspects" indicates the number of positive reviews in which at least one positive aspect is mentioned, while "#Neg Aspects" denotes the number of negative reviews that include at least one negative aspect.

We observe that in all three datasets, over 97% of the positive reviews contain mentions of positive aspects, suggesting that users tend to express the strengths of an item when they are satisfied. Interestingly, only around 92% of the negative reviews include mentions of negative aspects, implying that some users provide low ratings without explicitly stating their reasons in the review text.

TABLE 4.4: Statistics of original reviews and aspects in each dataset.

Dataset	#Aspects	Positive Reviews			Negative Reviews		
		#Total	#Pos Aspects	Ratio	#Total	#Neg Aspects	Ratio
Office	10,686	52,040	51,327	98.63%	6,249	5,770	92.33%
Video Games	29,738	82,941	81,202	97.90%	20,683	19,428	93.93%
CDs & Vinyl	85,381	445,087	440,883	99.06%	82,239	75,774	92.14%

4.5.2 Aspect Filtering

After extracting aspects from the original reviews, we filtered out aspects whose total occurrences did not meet a threshold, retaining only the most relevant aspects. Rather than choosing a fixed threshold, we determined an appropriate threshold for each dataset. Specifically, we ranked all aspects (excluding "overall") by their frequency in the dataset, gradually removing the least frequent aspects (irrelevant aspects) and monitoring the proportion of total reviews covered by the remaining aspects. We then identified the frequency threshold at which coverage was closest to 80%. The thresholds and coverage ratios identified for the Office, Video Games, and CDs & Vinyl datasets were as follows: (75, 81.57%), (68, 83.98%) and (98, 89.43%). For example, in the Office dataset, we considered only aspects with no less than 75 occurrences as relevant, and these covered 81.57% of the reviews.

The statistics of the filtered aspects for each dataset are presented in Table 4.5, excluding the generic aspect "overall". We observe a significant reduction in the number of aspects after filtering across all datasets. For example, the original Video Games dataset contains 29,738 unique aspects, which are reduced to 661 after filtering, while still covering 83.98% of the reviews. This filtering process not only removes noise from infrequent aspects but also alleviates the sparsity of the aspect space. The top 10 filtered aspects for each dataset are listed in Table 4.2, and they are highly representative of the domain-specific content of their respective datasets.

TABLE 4.5: Statistics of filtered aspects in each dataset.

Dataset	#Aspects	User			Item		
		#Max	#Min	#Ave	#Max	#Min	#Ave
Office	301	150	1	26.7	91	2	27.5
Video Games	661	439	1	40.0	235	4	51.0
CDs & Vinyl	1,304	804	1	43.6	429	4	56.6

4.5.3 Aspect Embedding Generation

When generating aspect embeddings, some aspects are not explicitly mentioned in the reviews, making it infeasible to use models like Word2Vec to derive embeddings by aggregating word representations. To address this, we used pre-trained word vectors

(Mikolov et al., 2018) from fasttext-wiki-news-subwords-300⁹ to create aspect embeddings directly. Specifically, for any given aspect, we extracted each word it contains, retrieved the corresponding vector from the pre-trained word vectors, and calculated the average to obtain the aspect’s embedding. Certain challenging aspects across the three datasets had no corresponding vectors; for these cases, we applied manual adjustments to ensure successful extraction. The challenging aspects and our adjustments are as follows:

TABLE 4.6: Challenging aspects and their adjustments.

Challenging Aspects	Adjusted Aspects
erasability	erase ability
collector\’s	collector
coltrane	jazz musician
sacd	super audio CD
listenability	audio quality
r&b	rhythm blues
80\’s	1980s

4.5.4 Baselines and Parameter Settings

Our model tackles two tasks: rating prediction and recommendation explanation, operationalised in this study as positive-aspect prediction. Accordingly, we evaluated DRL against baselines for the two tasks separately. For the explanation task, direct baselines were selected to match the output and evaluation setting of DRL. Specifically, the compared methods model aspect-level user preferences and item characteristics and produce supportive aspect-level explanations for an individual user-item pair within the same aspect space, allowing their outputs to be evaluated against the same positive aspects appearing in held-out reviews.

Among the selected baselines, EFM (Zhang et al., 2014a) and HDE (Luo et al., 2019) support both rating prediction and recommendation explanation and were therefore compared with DRL on both tasks. PMF (Mnih and Salakhutdinov, 2008) and AARM (Guan et al., 2019) were used only for rating prediction because they do not provide explanation outputs compatible with the current evaluation protocol. A simple Matching baseline was also included for the explanation task because it directly combines

⁹<https://fasttext.cc/docs/en/english-vectors.html>

historical user-aspect attention and item-aspect quality within the same aspect space. The following sections describe the characteristics of the baselines and their parameter settings.

Rating Prediction Baselines We evaluated our proposed model, DRL, against the following methods in the rating prediction task:

- **PMF** (Mnih and Salakhutdinov, 2008): Probabilistic Matrix Factorisation (PMF) is a classic probabilistic factor-based model that approximates the rating matrix by decomposing it into the product of two lower-rank matrices for users and items.
- **EFM** (Zhang et al., 2014a): The Explicit Factor Model (EFM) is a hybrid matrix factorisation approach that integrates the user-aspect attention matrix X and the item-aspect quality matrix Y , alongside the overall user-item rating matrix R . Specifically, X and Y share the same aspect latent factors, while the user and item latent factors that each decompose are subsets of the user and item latent factors derived from R .
- **AARM** (Guan et al., 2019): The Attentive Aspect-based Recommendation Model (AARM) represents users and items using semantic embeddings of aspects extracted from their respective reviews. To capture the connections among aspects (such as synonymous and similar aspects), it explores bi-interactions between these aspect embeddings, thereby enhancing the modelling of user preferences for items. Please note that AARM is originally designed for the top-N recommendation task. To enable comparison on the rating prediction task, we modified its loss function to MSE.
- **HDE** (Luo et al., 2019): The Hybrid Deep Embedding (HDE) model captures the time-evolving representations of users and items by learning semantic embeddings from their respective review sequences. During training, it employs the user-aspect attention vector x and item-aspect quality vector y as supervisory signals for the review-level representations.

Explanation Baselines For the recommendation explanation task, all methods first model the user’s attention and the item’s quality across all aspects, then select the aspects with the highest values derived from the element-wise multiplication of these representations. The baselines are as follows:

- **Matching:** This method directly uses the user-aspect attention vector $\mathbf{x}$ and item-aspect quality vector $\mathbf{y}$ extracted from the training set.
- **EFM** (Zhang et al., 2014a): EFM employs matrix factorisation to derive the user-aspect attention vector $\hat{\mathbf{x}}$ and item-aspect quality vector $\hat{\mathbf{y}}$ by learning the latent factors of users, items, and aspects.
- **HDE** (Luo et al., 2019): HDE first models the review-level representations of users and items, then transforms these representations into the user-aspect attention vector $\hat{\mathbf{x}}$ and item-aspect quality vector $\hat{\mathbf{y}}$, respectively.

Parameter Settings Hyperparameters were tuned using the validation set. The model exhibiting the lowest mean squared error (MSE) on the validation set was selected as the best-performing configuration and subsequently evaluated on the test set. For all models, the embedding dimension and batch size were fixed at 64 and 1024, respectively. A grid search was conducted for hyperparameter optimisation: the learning rate was selected from $\{1e-5, 1e-4, 1e-3, 1e-2\}$, and the L_2 regularisation coefficient was chosen from $\{1e-6, 1e-5, 1e-4, 1e-3\}$. For the explanation component in our model, as well as in EFM and HDE, the explanation weight was tuned within the range $[0.05, 1.0]$ using a step size of 0.05. Dropout rates were set according to those reported in the original publications: 0.5 for AARM and 0.3 for HDE. For our model, we adopted a dropout rate of 0.5. An early stopping strategy was employed, terminating training if the validation MSE did not improve over 50 consecutive epochs.

4.5.5 Rating Prediction Evaluation

Table 4.7 presents the performance of each model on the rating prediction task across the three datasets. Our observations are as follows:

- Our proposed model, DRL, consistently outperformed all baselines in terms of MSE and MAE. Notably, DRL demonstrated superior performance on denser datasets; for instance, it improved rating prediction accuracy by approximately 1.5% over the strongest baseline, HDE, on the sparsest dataset, CDs & Vinyl, while showing a significant 7% improvement on the densest dataset, Office.
- PMF, which relies solely on rating information and traditional matrix factorisation, performed less effectively than models that incorporate review information or use neural network techniques.

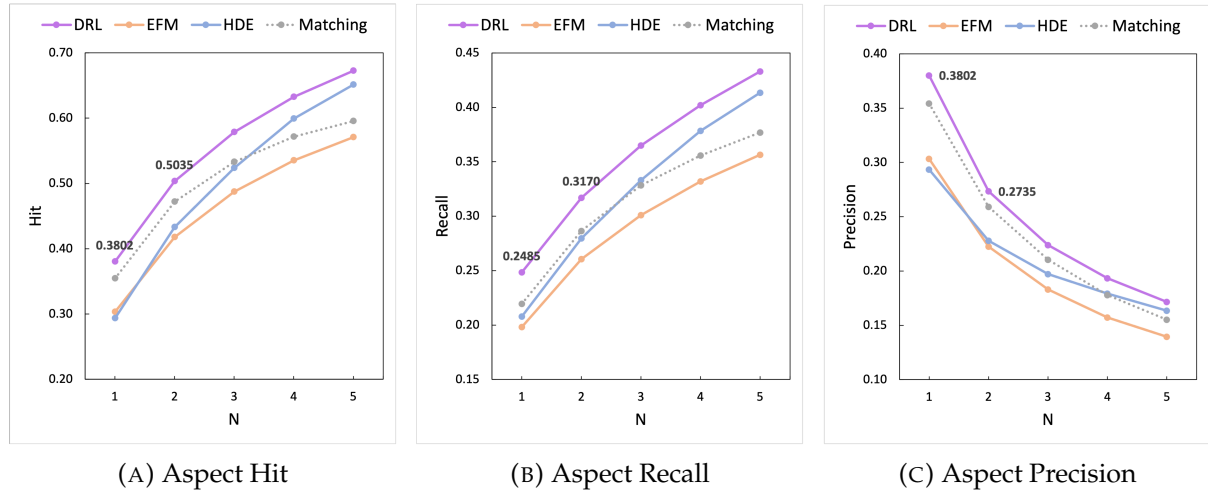
- EFM, which leverages aspect-level review information, surpassed PMF on MSE across all three datasets, with improvements of 8.61%, 6.54%, and 3.62%, respectively. However, EFM’s performance was lower than models employing neural networks, such as AARM, HDE, and our model, DRL.
- AARM significantly outperformed the two matrix factorisation models in rating accuracy, as it captures semantic relationships among aspects. However, its performance was lower than methods that incorporate sentiment information, such as HDE and our model, DRL.
- Among the baselines, HDE achieved the best overall performance. Compared to the second-strongest baseline, AARM, it reduced MSE by 2.75%, 14.35%, and 7.67%, and MAE by 4.26%, 9.94%, and 6.06% on the Office, Video Games, and CDs & Vinyl datasets, respectively.

TABLE 4.7: Overall comparison of rating prediction performance. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Office		Video Games		CDs & Vinyl	
Method	MSE	MAE	MSE	MAE	MSE	MAE
PMF	1.80649	0.96250	1.65110	1.00316	1.11809	0.82415
EFM	1.65097	0.91154	1.54316	0.97203	1.07767	0.81105
AARM	0.65345	0.57557	1.20564	0.80546	0.82945	0.66268
HDE	<u>0.63548</u>	<u>0.55106</u>	<u>1.03262</u>	<u>0.72538</u>	<u>0.76594</u>	<u>0.62253</u>
DRL	0.58603	0.51245	0.97370	0.70665	0.75192	0.61285
%Improv.	7.78%	7.01%	5.70%	2.58%	1.83%	1.55%

4.5.6 Explainability Evaluation

We employed an offline formula-based approach to quantitatively assess whether the predicted positive aspects appeared in the ground-truth reviews, and conducted a case study to qualitatively evaluate the relevance and quality of presenting these aspects as recommendation explanations to users.

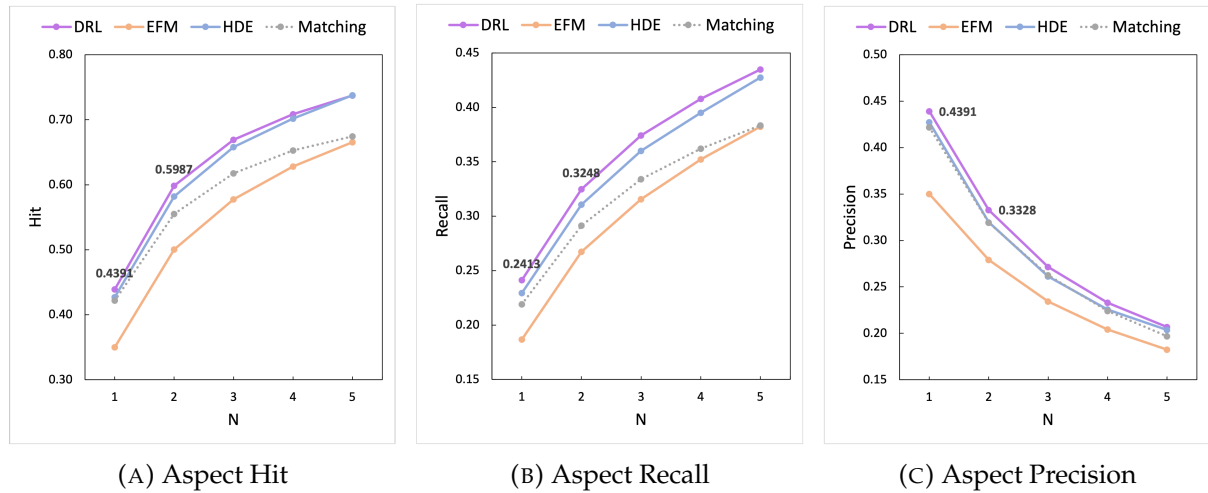


(A) Aspect Hit

(B) Aspect Recall

(C) Aspect Precision

FIGURE 4.4: Explainability performance on the **Office** dataset with varying numbers of predicted aspects. Subplots (A), (B), and (C) report Hit, Recall, and Precision respectively. For DRL, exact values at $N = 1$ and $N = 2$ are annotated on the curves.



(A) Aspect Hit

(B) Aspect Recall

(C) Aspect Precision

FIGURE 4.5: Explainability performance on the **Video Games** dataset with varying numbers of predicted aspects. Subplots (A), (B), and (C) report Hit, Recall, and Precision respectively. For DRL, exact values at $N = 1$ and $N = 2$ are annotated on the curves.

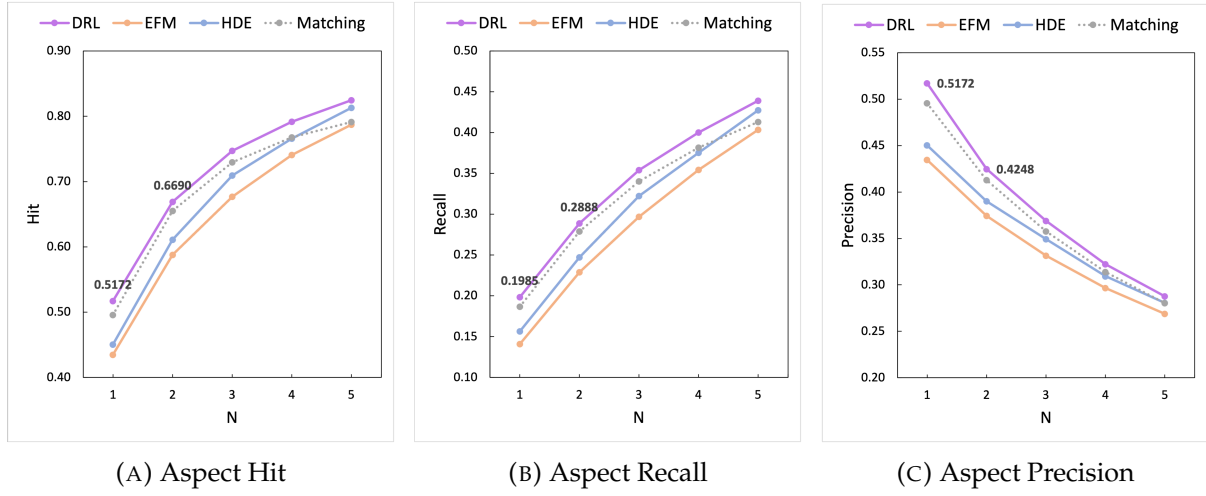


FIGURE 4.6: Explainability performance on the **CDs & Vinyl** dataset with varying numbers of predicted aspects. Subplots (A), (B), and (C) report Hit, Recall, and Precision respectively. For DRL, exact values at $N = 1$ and $N = 2$ are annotated on the curves.

Offline Formula-Based Method

We utilised three metrics for evaluating model-generated explanations: Hit@ N , Recall@ N , and Precision@ N . Higher values of these metrics indicate greater accuracy in predicting aspects that users find satisfactory. We vary N from 1 to 5, thereby computing each metric when selecting between 1 and 5 aspects as explanatory outputs. We argue that aspects ranked higher are of greater importance, given users' limited attention span and the typical constraints on explanation display space. As such, the values of these metrics at $N = 1$ are considered more indicative of a model's explanatory precision than those at higher N .

In Section 4.4.2, we introduced two settings based on different review selection criteria: reviews associated with actual ratings greater than 3 versus those associated with predicted ratings greater than 3. The former evaluates aspect prediction against held-out positive review evidence, whereas the latter restricts evaluation to instances that the model itself would recommend based on its predicted rating.

We first report explanation performance using reviews associated with actual ratings greater than 3 as the reference. To better illustrate the comparative results across models, we plotted line charts for each dataset, with N on the x-axis and the corresponding metric values on the y-axis. The visual results for the Office, Video Games,

and CDs & Vinyl datasets are presented in Figures 4.4, 4.5, and 4.6, respectively. Complementing the visualisations, Table 4.8 reports the detailed metric values¹⁰ across the three datasets for various values of N , with all metric values presented as percentages. Note that, by definition, Hit@1 and Precision@1 are equivalent.

TABLE 4.8: Overall comparison of explanation performance. In this evaluation, the ground truth consists of reviews associated with **actual ratings** greater than 3. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Office					Video Games					CDs & Vinyl				
Method	Ht@1	Ht@2	Ht@3	Ht@4	Ht@5	Ht@1	Ht@2	Ht@3	Ht@4	Ht@5	Ht@1	Ht@2	Ht@3	Ht@4	Ht@5
Matching	<u>35.44</u>	47.19	53.29	57.17	59.54	42.18	55.53	61.77	65.29	67.46	<u>49.56</u>	<u>65.51</u>	<u>72.94</u>	76.79	79.15
EFM	30.33	41.77	48.73	53.53	57.09	35.01	50.05	57.77	62.81	66.56	43.47	58.78	67.70	74.09	78.74
HDE	29.35	43.29	52.35	<u>59.91</u>	<u>65.12</u>	<u>42.73</u>	58.20	<u>65.81</u>	70.17	73.78	45.05	61.10	70.95	76.61	<u>81.32</u>
DRL	38.02	50.35	57.88	63.26	67.26	43.91	59.87	66.91	70.87	<u>73.72</u>	51.72	66.90	74.72	79.18	82.45
%Improv.	7.28%	6.70%	8.61%	5.59%	3.29%	2.76%	2.87%	1.67%	1.00%	-0.08%	4.36%	2.12%	2.44%	3.11%	1.39%
Method	Re@1	Re@2	Re@3	Re@4	Re@5	Re@1	Re@2	Re@3	Re@4	Re@5	Re@1	Re@2	Re@3	Re@4	Re@5
Matching	<u>21.96</u>	<u>28.66</u>	32.83	35.58	37.68	21.92	29.14	33.40	36.22	38.35	<u>18.68</u>	<u>27.89</u>	<u>34.02</u>	<u>38.17</u>	41.30
EFM	19.83	26.06	30.11	33.20	35.64	18.67	26.74	31.57	35.22	38.22	14.10	22.91	29.70	35.45	40.36
HDE	20.79	27.98	<u>33.32</u>	<u>37.85</u>	<u>41.35</u>	<u>22.94</u>	<u>31.06</u>	<u>36.01</u>	<u>39.50</u>	<u>42.75</u>	15.64	24.71	32.23	37.53	<u>42.73</u>
DRL	24.85	31.70	36.52	40.21	43.31	24.13	32.48	37.42	40.79	43.49	19.85	28.88	35.42	40.02	43.90
%Improv.	13.16%	10.61%	9.60%	6.24%	4.74%	5.19%	4.57%	3.92%	3.26%	1.73%	6.26%	3.55%	4.12%	4.85%	2.74%
Method	Pr@1	Pr@2	Pr@3	Pr@4	Pr@5	Pr@1	Pr@2	Pr@3	Pr@4	Pr@5	Pr@1	Pr@2	Pr@3	Pr@4	Pr@5
Matching	<u>35.44</u>	<u>25.91</u>	<u>21.03</u>	17.78	15.51	42.18	31.90	<u>26.26</u>	22.40	19.68	<u>49.56</u>	<u>41.30</u>	<u>35.78</u>	<u>31.40</u>	28.02
EFM	30.33	22.25	18.30	15.72	13.94	35.01	27.91	23.43	20.42	18.24	43.47	37.44	33.15	29.65	26.89
HDE	29.35	22.78	19.73	<u>17.92</u>	<u>16.35</u>	<u>42.73</u>	<u>31.96</u>	26.11	<u>22.58</u>	<u>20.36</u>	45.05	39.01	34.92	30.93	<u>28.06</u>
DRL	38.02	27.35	22.39	19.34	17.16	43.91	33.28	27.14	23.30	20.67	51.72	42.48	36.88	32.23	28.77
%Improv.	7.28%	5.56%	6.47%	7.92%	4.95%	2.76%	4.13%	3.35%	3.19%	1.52%	4.36%	2.86%	3.07%	2.64%	2.53%

Our observations on the models' explanation performance are as follows:

- Our proposed model, DRL, outperformed all baselines across all metrics and datasets, except for Hit@5 on the Video Games dataset. When selecting only one aspect as an explanation, DRL showed the largest improvement over other baselines. More specifically, on the Office, Video Games, and CDs & Vinyl datasets, it achieved increases of 7.28%, 2.76%, and 4.36% on Hit/Precision@1, respectively, and improved Recall@1 by 13.16%, 5.19%, and 6.26%. Overall, our model showed a particularly strong performance in Recall, indicating higher recall of positive aspects appearing in the held-out reviews. Furthermore, DRL demonstrated larger gains over the baselines on the Office and CDs & Vinyl datasets compared to Video Games. We attributed this to specific characteristics of the Video Games dataset. As shown in Table 4.4, the proportion of positive aspects mentioned in

¹⁰Due to space limitations, we abbreviate Hit@ N , Recall@ N , and Precision@ N as Ht@ N , Re@ N , and Pr@ N , respectively.

reviews with ratings above 3 is lowest for Video Games at 97.90%, compared to 98.63% and 99.06% for Office and CDs & Vinyl, respectively. This suggests that some users give high ratings in Video Games without explicitly mentioning positive aspects.

- Surprisingly, Matching, which uses a straightforward approach by directly employing the user-aspect attention vector x and item-aspect quality vector y extracted from the training set, achieves reasonable explainability. Although its performance on Video Games did not surpass HDE, it stands as the strongest baseline across most metrics on Office and CDs & Vinyl, especially with lower N values.
- EFM employs matrix factorisation to infer the unobserved (zero) values in the user-aspect attention vector x and item-aspect quality vector y , thus learning $\hat{x}$ and $\hat{y}$ to generate recommendation explanations. However, this approach did not perform as well as the simpler Matching method. This may be due to the limitations of matrix factorisation, which can only capture shallow features.
- HDE exhibited solid explainability performance on Video Games but was less competitive on Office and CDs & Vinyl. A clear trend for HDE was that as the number of aspects in the explanation increases, its performance on all three metrics improved. This may be because HDE processes full review content rather than isolated aspects and their sentiments, capturing more comprehensive information at the cost of potential noise. Consequently, when only a limited number of aspects are allowed in the explanation, the predictions may lack accuracy. However, as the number of aspects in the explanation expands, the advantage of encoding full review content becomes apparent, leading to stronger performance.

As discussed in Section 4.4.2, in practical scenarios, the model only recommends an item and provides explanations if the predicted rating exceeds 3. Accordingly, we also report explanation performance based on reviews of items with predicted ratings above 3 (see Table 4.9). Since the Matching method does not involve rating prediction, its results are not available under this evaluation setting.

Overall, compared to the setting that uses reviews of items with actual ratings above 3, all models showed a decline in explanation performance. We speculate that this is because aspect prediction in this setting is more sensitive to rating prediction accuracy. For example, if a user actually dislikes an item (e.g., gives a rating of 1), but the

TABLE 4.9: Overall comparison of explanation performance. In this evaluation, the ground truth consists of reviews associated with **predicted ratings** greater than 3. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Office					Video Games					CDs & Vinyl				
Method	Ht@1	Ht@2	Ht@3	Ht@4	Ht@5	Ht@1	Ht@2	Ht@3	Ht@4	Ht@5	Ht@1	Ht@2	Ht@3	Ht@4	Ht@5
EFM	<u>29.68</u>	<u>40.86</u>	47.38	51.91	55.25	32.05	46.20	53.66	58.38	61.92	40.06	54.87	63.78	69.94	74.41
HDE	27.51	40.63	<u>49.16</u>	<u>56.31</u>	<u>61.30</u>	<u>37.14</u>	<u>51.42</u>	<u>58.69</u>	<u>63.04</u>	66.62	<u>41.28</u>	<u>56.62</u>	<u>66.25</u>	<u>71.71</u>	<u>76.18</u>
DRL	35.45	47.11	54.24	59.38	63.16	38.21	52.78	59.63	63.46	66.49	47.00	61.55	69.29	73.68	76.85
%Improv.	19.44%	15.30%	10.33%	5.45%	3.03%	2.88%	2.64%	1.60%	0.67%	-0.20%	13.86%	8.71%	4.59%	2.75%	0.88%
Method	Re@1	Re@2	Re@3	Re@4	Re@5	Re@1	Re@2	Re@3	Re@4	Re@5	Re@1	Re@2	Re@3	Re@4	Re@5
EFM	19.84	26.08	29.98	32.99	35.32	17.31	24.96	29.65	33.06	35.88	13.27	21.82	28.53	33.99	38.64
HDE	19.51	<u>26.30</u>	<u>31.35</u>	<u>35.68</u>	<u>39.04</u>	<u>20.00</u>	<u>27.58</u>	<u>32.30</u>	<u>35.73</u>	<u>38.87</u>	<u>14.62</u>	<u>23.34</u>	<u>30.62</u>	<u>35.66</u>	<u>40.50</u>
DRL	23.25	29.75	34.32	37.84	40.79	20.98	28.69	33.40	36.61	39.34	18.30	26.96	33.30	37.71	41.34
%Improv.	17.19%	13.12%	9.47%	6.05%	4.48%	4.90%	4.02%	3.40%	2.46%	1.21%	25.17%	15.51%	8.75%	5.75%	2.07%
Method	Pr@1	Pr@2	Pr@3	Pr@4	Pr@5	Pr@1	Pr@2	Pr@3	Pr@4	Pr@5	Pr@1	Pr@2	Pr@3	Pr@4	Pr@5
EFM	<u>29.68</u>	21.71	17.73	15.20	13.43	32.05	25.70	21.73	18.94	16.93	40.06	34.71	30.88	27.61	25.04
HDE	27.51	21.35	<u>18.46</u>	<u>16.78</u>	<u>15.32</u>	<u>37.14</u>	<u>28.18</u>	<u>23.15</u>	<u>20.10</u>	<u>18.16</u>	<u>41.28</u>	<u>35.88</u>	<u>32.22</u>	<u>28.52</u>	<u>25.85</u>
DRL	35.45	25.55	20.92	18.09	16.05	38.21	29.28	24.04	20.70	18.44	47.00	38.81	33.78	29.52	26.36
%Improv.	19.44%	17.69%	13.33%	7.81%	4.76%	2.88%	3.90%	3.84%	2.98%	1.54%	13.86%	8.17%	4.84%	3.51%	1.97%

model predicts a rating of 4, it is then required to identify aspects the user supposedly appreciated (an explanation that is ultimately meaningless). Conversely, if a truly liked item is predicted to have a low rating, the corresponding review is excluded from the evaluation, and the model's prediction is not assessed. Both scenarios introduce noise due to prediction errors.

Nevertheless, the relative performance ranking among models remained consistent with the setting based on actual ratings. Specifically, DRL continued to deliver the strongest explanation performance. EFM performed well only at Hit@1 on the Office dataset, but was outperformed by HDE across other metrics. HDE, in turn, showed substantial gains in aspect prediction N increases.

Case Study

Tables 4.10, 4.11, and 4.12 present three illustrative cases from the Office, Video Games, and CDs & Vinyl datasets, respectively. Each case includes the user & item ID (Column 1), the actual rating (Column 2), and the predicted rating (Column 3) by DRL. The top-5 positive aspects predicted by DRL are listed in Column 4. Aspects that are correctly matched, i.e., those appearing in the ground-truth review, are highlighted in

blue within the actual review text (Column 6)¹¹. Column 5 presents the filtered aspects extracted from the review, with positive aspects shown in blue and negative aspects in red.

TABLE 4.10: Predicted rating and top-5 positive aspects versus actual rating and user review (**Office Dataset**).

User ID Item ID	Actual Rating	Predicted Rating	Top-5 Predicted Positive Aspects	Extracted Aspects	Actual Review
A1PEYX9SUVT874 B00O32UHHHC	5	4.80	price quality color durability size	color texture size quality price shipping durability	"I ordered this desk chair pad to keep my rollator from wearing out the carpet and because of the color. It also hides the stains from my many spills. Nice texture and just the right size. It appears to be of a better quality than other pads I have bought. Good price and excellent shipping. Glad I ordered it and would buy again. It is not stiff and came flat. Will last me a long time."
A2ZJPDH0YP90SD B01H9SH2OW	5	4.47	design color durability appearance quality	appearance color design durability	"These are really gorgeous dividers. Brilliant, bright colors on one side and cool black and white patterns on the other. Each divider page is a different bright color and the back is a different black and white design. As you can see in the pics I posted - you get five tab pages, thick and will probably last all year. The one downside is that the tab labels that come with these cover the pretty colors that stick out from the black and white patterns. So I just used a black sharpie to write in the subjects. You can use the labels with an inkjet or laser printer, but they will cover the bright colors. The labels are plain white."
A3LCT5601UA4UB B00A89YZ6S	5	4.66	price quality delivery value overall	quality price delivery	"Good quality, price and on time delivery"

In the three Office cases (Table 4.10), the predicted ratings closely match the users' actual ratings. More importantly, the model successfully identifies aspects that users explicitly expressed satisfaction with. For instance, in the first case, the model predicts the top-5 positive aspects as "price", "quality", "color", "durability", and "size". Among these, "color", "size", "quality", and "price" are explicitly mentioned in the review. Although "durability" is not directly stated, the sentence "Will last me a long time" implies user satisfaction with that aspect, indicating the model's prediction is semantically valid. This case also highlights the capability of using LLM-based ABSA to infer implicitly expressed aspects.

¹¹To improve clarity, the original reviews were processed as follows: (1) Non-opinionated content (e.g., item descriptions) is shown in gray; (2) Overly long reviews are partially truncated, with omissions indicated by ...; and (3) Sensitive content is masked using ***.

TABLE 4.11: Predicted rating and top-5 positive aspects versus actual rating and user review (Video Games Dataset).

User ID Item ID	Actual Rating	Predicted Rating	Top-5 Predicted Positive Aspects	Extracted Aspects	Actual Review
A1PL6VA9OKD155 B00JK00S0S	5	4.51	gameplay graphic story game price	game graphic story gameplay	"This is one of the best games ever, the graphics are amazing, strong story, and the gameplay is amazing. This is the best horror survival game hands down, period."
A2K6AG0FYPXRC4 B017L186MY	5	5.04	delivery gift exercise condition description	delivery	"all good very fast"
A1IQV6AI7L9NOK B0002CHJAA	5	3.66	story gameplay graphic character music	graphic music character story battle system gameplay	"...Graphics (score 9/10): If you just finished the first game and then popped this one in, the first thing that you would notice is that the art style is different. It's not that it's bad or anything, but coming right after the first game, the difference is jarring. ...As I said, it's not bad, just a jarring change. Music (score 9/10): The music is just as beautiful as it was in the first game. Absolutely stunning tracks that lend great emotional depth to the scenes they are featured in. As for voice acting, all of it's good - well, with one exception: Momo's voice is really irritating. It's just high pitched and annoying. ...Characters (score 10/10): ...They are just as compelling as they were the first time, and they get a lot of great development too. ...I like that in this entry, you get a lot of back story on him which really helps flesh out his motivations as a character. Story (score 10/10): ...Suffice it to say, the story is very confusing if you haven't played the first game. It's even more complex than the first game's storyline, and features many disturbing and heart-wrenching scenes (perhaps even more than the first game). This game definitely focuses on Jr. and Albedo's backstory moreso than anything else, and I'm not unhappy with that as it's very interesting. ...Battle System (score 6/10): Well, combat's where everything kind of goes downhill. Battles are really irritating due to the break system. It seems like it takes ages to break an enemy, making battles really tedious. You're also now forced to bring Momo if you want a reasonable chance of success. ...So, basically, character balance is not fantastic in this regard. Couple that with extremely tedious battles where you've got to break every single enemy to do any decent damage, and you've got a big dent in the enjoyability of this game. In fact, out of all three games in the series, this entry has the least enjoyable battle system for me. Gameplay (score 10/10): This is a bit of a catch-all category for me. Well, menus, camera angles, all the standard stuff is fine. One thing of note here is that there are no random battles. ...On the other hand, the cutscenes in this game are just as long as unrelenting as the first game's. ...I, however, didn't mind at all. There is one thing left I should mention - this series continues the same button layout as the first game. ...It's backwards to most other PS2 games, and it can take some getting used to. Overall (score 9/10): ...The battle system of this game makes it feel very tedious at times, but the story, characters, and music are still fantastic. Plus, you really need to play this one to get to the third game in the series, so I'd still recommend it despite it's flaws."

In the second and third cases, the number of extracted aspects from the ground-truth reviews is fewer than five; therefore, the model's top-5 predictions include aspects not directly mentioned in the text. Nonetheless, in the second case, the fifth predicted aspect, "quality", is semantically aligned with the expressed aspect "durability"; and in the third case, aspects such as "value" and "overall" correspond meaningfully to the review's expressed concerns, including "quality", "price", and "delivery". Additionally, the second case illustrates the noisy nature (segments lacking sentiment information) of user reviews, which poses challenges for aspect-level extraction and highlights the importance of robust ABSA techniques.

Table 4.11 shows three cases from the Video Games dataset. The first two follow a pattern similar to those in the Office dataset, where both the predicted ratings and the top-5 positive aspects align well with the actual ratings and reviews. The third case is particularly noteworthy: the user explicitly provides numeric ratings for each aspect, followed by detailed explanations. For example, the user gives a score of 9 out of 10 for "graphics", indicating satisfaction while acknowledging minor limitations. Among the six aspects mentioned (excluding "overall"), five ("graphics", "music", "character", "story", and "gameplay") receive high ratings, while "battle system" is rated lower (6/10).

Interestingly, ABSA identifies "battle system" as a negative aspect, likely inferred through contrast with the other, more highly rated aspects. This interpretation may have led the model to predict a lower overall rating of 3.66, significantly underestimating the user's actual rating of 5. This discrepancy suggests that the model inferred the user's dissatisfaction with "battle system" would have a stronger negative impact on the overall score. However, the user's final rating reflects a more generous attitude, possibly indicating a tendency to give high ratings despite some critical feedback.

Across the Office and Video Games cases, we observe that some extracted aspects are overly broad in meaning, for example, "game" in the Video Games dataset. This issue becomes even more pronounced in the CDs & Vinyl dataset (Table 4.12), where general aspects such as "album", "song", "track", "sound", and "music" are frequently extracted. While these broad aspects may be suboptimal for user-facing explanations, we argue that they nonetheless enhance model transparency by indicating which positive aspects contribute to higher ratings and which negative ones lower them. The first two cases in Table 4.12 demonstrate accurate predictions for both rating and aspects. The third case illustrates that when no specific aspects can be extracted from a review, our ABSA method assigns a generic "overall" aspect, yet the model can still predict it

TABLE 4.12: Predicted rating and top-5 positive aspects versus actual rating and user review (CDs & Vinyl Dataset).

User ID Item ID	Actual Rating	Predicted Rating	Top-5 Predicted Positive Aspects	Extracted Aspects	Actual Review
A2P1U2ACG187I8 B000002IYWQ	4	4.24	album song track guitar sound	guitar keyboard sound track song album drumming bass	"This was the follow-up to Roll the Bones and saw the band get a lot more into electrical guitar and less on the keyboards and synthesizers to back up the sound. Standout tracks include "Stick it Out", the instrumental "Leave that Thing Alone", and "Everyday Glory." Though probably not as popular as Rush's earlier stuff, from the mid to late 80's, this is a solid album all the way through and displays what the band does best, write good songs backed up by nice guitar, bass, excellent drumming and some keyboards."
A26XCS3QLDJJWN B00001IVOY	4	4.21	album track song lyric beat	album collaboration lyric voice delivery producer production track beat effect guest appearance song	"This is a really good album. Why did Xzibit have to h*** up with Dr. ****? Boo. On "At the Speed of Life," Xzibit was still warming his lyrical chops up, and there was definite talent there. He had the right voice, the right delivery, and DEFINITELY the right producers. And "40 Dayz..." is just the next step up. The production on this entire album is pretty tight, as X spreads the board duties around. Producers as diverse as Soopafly, Montageone and E-Swift make contributions to the disc, all of them tight. For highlights, take a gander at the following: "Chamber Music" samples the beat from Queen's "We Will Rock You" and flips it with some ominous strings for a vicious opener. The heavy bounce of "What You See Is What You Get," "Nobody Sound Like Me," and "Los Angeles Times" will keep you moving, and the Likwit Crew cut "Let It Rain" hits like a 20-lb. sledge, culminating in the syncopated interplay of the melancholy "Recycled Assassins." ..., but it's just so much fun to listen to, as Xzibit swaps verses with Jayo Felony (in what I must say is a very good guest appearance). Method Man even shows up on the hook, literally. Apparently he just showed in the studio and asked if he could do the hook. And BAM, great song. There are some flatliners, though. "Shroomz" plods along, and "Handle Your Business" is a little boring, but they don't detract too much from the overall good vibe. This is a really good album. I just hope "The Restless LP" doesn't ruin Xzibit's underground credibility, which is the basis for his success thus far. ..."
AMU77WVN01U9B B00000DMUL	5	4.71	overall music album sound collection	overall	"wow, loved it"

accurately.

In summary, these case studies illustrate that the predicted aspects can correspond closely to aspect-level evidence in users' held-out reviews. They provide qualitative examples of how the model's outputs can be associated with identifiable item characteristics, but do not establish user-perceived explanation quality.

4.5.7 Effect of Varying GNN Layer Depths

We varied the depth of GNN layers in the DRL model and reported its performance in both rating prediction and aspect prediction at different layers. Figure 4.7 illustrates the rating prediction results across the three datasets, with the x-axis representing the total number of convolutional layers and the dual y-axes showing MSE on the left and MAE on the right. Since the best model was selected based on the lowest MSE, DRL achieved optimal performance at two layers on the Office and CDs & Vinyl datasets, even though the lowest MAE occurred at a single layer in these datasets. On the Video Games dataset, both MSE and MAE were minimised at one layer. Overall, DRL performed best in rating prediction with either one or two GNN layers, and increasing the depth to three did not further enhance recommendation accuracy.

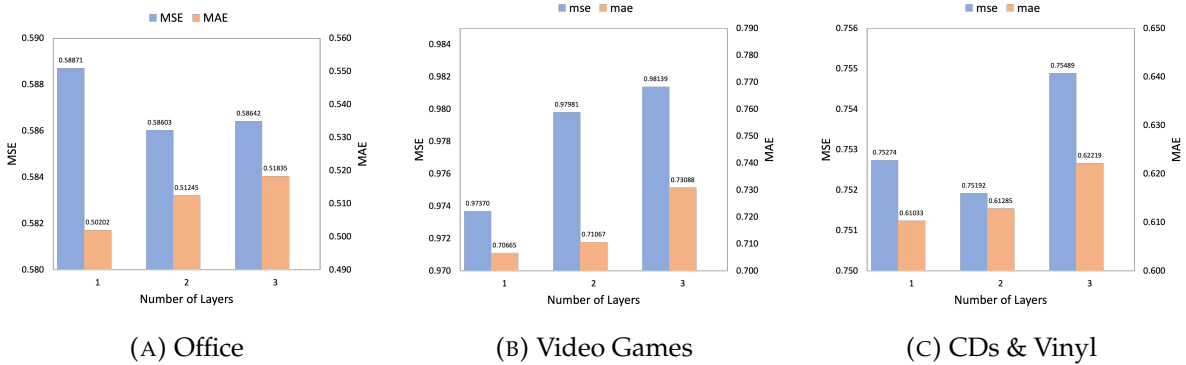


FIGURE 4.7: Rating prediction performance of DRL at different GNN layers across different datasets.

Table 4.13 presents the Hit@1 and Recall@1 scores for aspect prediction across different GNN layers on each dataset. Consistent with the rating prediction findings, DRL generally achieved the highest explainability performance with two GNN layers.

TABLE 4.13: Explainability performance of DRL at different GNN layers across different datasets.

Dataset	Office		Video Games		CDs & Vinyl	
Method	Hit@1	Recall@1	Hit@1	Recall@1	Hit@1	Recall@1
1 Layer	33.83	22.86	43.91	24.13	50.99	19.16
2 Layers	38.02	24.85	43.87	24.21	51.72	19.85
3 Layers	37.93	24.53	43.17	24.08	50.61	19.27

4.5.8 Ablation Study

To validate the effectiveness of various components in our model, we conducted an ablation study by systematically removing specific parts and comparing their rating prediction and aspect prediction performance with the original model. The ablation study focuses on four components: the GNN layer, the target edge in the Dynamic Representation Layer, semantic aspect embeddings, and the rating as a dummy aspect. Detailed descriptions of each component’s ablation are provided in the following sections.

Remove GNN Layer

To examine the contribution of the GNN layer, we removed it and replaced the user and item embeddings output by the GNN layer with learnable user and item embeddings. This modified model is named DRL-gnn.

Retain Target Edge in Dynamic Representation Learning

When modelling the interaction-dependent user embedding and item embedding, to simulate the realistic interaction scenario, we excluded the target edge (review and rating) from the attention network input. In the DRL+target edge variant, we retained this edge and evaluated its impact on recommendation and explanation performance.

Disable Semantic Aspect Embeddings

The original model captured semantic connections among aspects by multiplying the sentiment-based aspect vector $\hat{a}_{ij} = [a_{ij1}, a_{ij2}, \dots, a_{ijp}, r_{ij}]$ with aspect embeddings $[W_A || w_d]$ within the edge encoder (Section 4.3.2) module. To test the effectiveness of incorporating semantic aspect embeddings, we removed this embedding layer and

directly mapped $\hat{a}_{ij}$ to the target space, bypassing the semantic embedding multiplication. This variant is referred to as DRL-semantic.

Remove Dummy Aspect

In the DRL-dummy model, we removed the dummy aspect by only multiplying $a_{ij} = [a_{ij1}, a_{ij2}, \dots, a_{ijp}]$ with W_A . Consequently, since the rating was no longer encoded as a dummy aspect, we did not reserve the final position for the rating in the prediction layer (Section 4.3.3) when predicting the user attention vector and item quality vector. Instead, we directly converted them to vectors of length p .

For each of the four model variants, we maintained all hyper-parameters (i.e., learning rate, regularisation coefficient, and dropout ratio) consistent with the optimal settings of DRL, while separately tuning the GNN layer count (if applicable) and the weight of the explanation component.

Table 4.14 presents the rating prediction performance of the original model, DRL, and the four variants. We observed that DRL-gnn exhibited higher MSE and MAE values on the Office and Video Games datasets compared to DRL, indicating that the GNN Layer positively contributes to the rating prediction accuracy. Additionally, removing the target edge in the Dynamic Representation Layer resulted in a slight improvement in recommendation accuracy on these two datasets. Surprisingly, on the CDs & Vinyl dataset, DRL-gnn and DRL+target edge outperformed DRL on the MSE and MAE metrics, respectively, even though their performance on the other metric was inferior to that of DRL. Moreover, excluding the dummy aspect had the most significant negative impact on recommendation performance. However, disabling all semantic aspect embeddings had the least impact on recommendation accuracy and, in half of the cases, even yielded positive effects; specifically, the MAE on the Video Games dataset was lower than that of DRL, as were both the MSE and MAE on the CDs & Vinyl dataset. Overall, the original model DRL demonstrated the strongest performance in the rating prediction task across all three datasets, with DRL-semantic closely following as the second-best performer.

The results of the ablation studies on aspect prediction, including Hit@1 and Recall@1 values, are provided in Table 4.15. The ablation analysis reports Hit@1 and Recall@1 because it focuses on the single highest-ranked predicted aspect. The effect of varying the number of predicted aspects is analysed separately in Section 4.5.6, where Hit@N, Recall@N, and Precision@N are reported for $N = 1, \dots, 5$. Consistent with the findings from the rating prediction task, the GNN Layer proved beneficial for aspect

TABLE 4.14: Rating prediction performance of DRL and its four variants. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Office		Video Games		CDs & Vinyl	
Method	MSE	MAE	MSE	MAE	MSE	MAE
DRL	0.58603	0.51245	0.97370	<u>0.70665</u>	0.75192	0.61285
DRL-gnn	0.59361	0.53336	0.98762	0.73316	0.74557	0.61326
DRL+target edge	0.58984	0.51992	0.97824	0.71070	0.75727	0.60263
DRL-semantic	<u>0.58868</u>	<u>0.51984</u>	<u>0.97635</u>	0.70463	<u>0.74664</u>	<u>0.61103</u>
DRL-dummy	0.60110	0.52967	0.99326	0.72018	0.76238	0.62459

prediction as well. After retaining the target edge or removing the dummy aspect, the model’s explainability decreased on the Video Games and CDs & Vinyl datasets but surprisingly improved on the Office dataset. Interestingly, disabling semantic aspect embeddings even boosted explainability on all three datasets.

TABLE 4.15: Explainability performance of DRL and its four variants. The best results are highlighted in bold, and the second-best results are underlined.

Dataset	Office		Video Games		CDs & Vinyl	
Method	Hit@1	Recall@1	Hit@1	Recall@1	Hit@1	Recall@1
DRL	38.02	24.85	<u>43.91</u>	<u>24.13</u>	<u>51.72</u>	<u>19.85</u>
DRL-gnn	31.65	22.45	41.61	23.60	47.67	16.59
DRL+target edge	38.84	24.95	43.31	24.02	49.41	17.78
DRL-semantic	39.95	25.28	44.86	24.54	53.50	20.29
DRL-dummy	<u>39.37</u>	<u>25.25</u>	42.39	23.75	46.08	14.60

4.6 Discussion

This study investigated the research question: **How can user and item representations be conditioned on each other to support both rating prediction and aspect-level recommendation explanation?** To address this question, DRL separated general collaborative information from interaction-specific information when constructing user and item representations. The rating-based GNN first learned general user and item embeddings from historical ratings and their collaborative neighbourhoods. For a target

user-item pair, the dynamic representation layer then incorporated aspect-sentiment information associated with their interaction, allowing the user representation to depend on the target item and the item representation to depend on the target user. In this study, "dynamic" therefore referred specifically to *interaction-dependent* representations rather than representations that evolved over time. The resulting representations were used for both rating prediction and the prediction of user-aspect attention and item-aspect quality.

The idea that user preferences can depend on the recommendation target is related to existing target-aware recommendation research (He et al., 2018b; Cao et al., 2023; Kim et al., 2024). Such approaches have shown that representing a user identically for every candidate item can overlook the parts of the user's historical preferences that are most relevant to the current target. DRL followed this broader motivation but investigated it in a different setting. Specifically, it conditioned both sides of the user-item interaction on the other entity in the target pair and incorporated review-derived aspect-sentiment information into this process. Its contribution was therefore not the general idea that a user representation can depend on the recommendation target, but a particular mechanism for combining collaborative and fine-grained review information to construct interaction-dependent user and item representations for both rating and aspect prediction.

The rating prediction results provided evidence that this representation strategy was beneficial under the evaluated setting. DRL reduced MSE relative to all compared baselines across the three datasets. Compared with HDE (Luo et al., 2019), which modelled time-evolving review representations and direct neighbour information, DRL additionally incorporated high-order collaborative information through the rating-based GNN and conditioned the final user and item representations on the current interaction. The experimental results were consistent with the hypothesis that combining general collaborative information with interaction-specific information could provide a more informative representation for predicting a user's evaluation of a particular item. Nevertheless, because these components interacted within the overall architecture, the baseline comparison alone could not establish which design element was solely responsible for the improvement.

The comparison with AARM (Guan et al., 2019) also provided some evidence for the usefulness of sentiment information. AARM represented users and items through review aspects without explicitly modelling their associated sentiment polarities, whereas

DRL incorporated aspect-sentiment information. AARM achieved lower rating prediction accuracy in the experiments. This result was consistent with the intuition that knowing whether an aspect was discussed positively or negatively can provide more informative preference evidence than knowing only that the aspect was mentioned. The comparison did not, however, isolate sentiment as the only difference between the two models, and therefore should not be interpreted as a direct causal demonstration of the benefit of sentiment information.

The explanation experiments offered a complementary finding. The Matching baseline directly combined the user-aspect attention vector and item-aspect quality vector obtained from historical data and performed surprisingly strongly, outperforming EFM (Zhang et al., 2014a). This finding suggested that the basic principle underlying aspect-level explanation was itself effective: an aspect was a plausible positive explanation when it was important to the user and evaluated positively for the item. DRL improved upon this static matching strategy by predicting interaction-dependent user attention and item quality rather than relying only on historical profiles. Under the evaluation protocol used in this study, this produced higher positive-aspect prediction performance than the compared explanation baselines.

The ablation experiments further clarified the contribution of individual model components. Removing the rating-based GNN reduced aspect prediction performance across all three datasets and generally weakened rating prediction, although the effect on rating prediction was not uniform across every dataset and metric. This suggested that high-order collaborative information provided useful general representations, particularly for predicting the aspects associated with a target interaction. The target-edge and dummy-aspect variants produced more dataset-dependent effects on aspect prediction. For rating prediction, the original DRL design generally performed better than these variants, indicating that removing the target edge and retaining the dummy rating aspect were useful design choices within the current architecture. Their effects on aspect prediction were less consistent, suggesting that the optimal treatment of these components may depend on dataset characteristics and prediction objectives.

The semantic aspect embeddings produced a particularly informative ablation result. Their effect on rating prediction varied across datasets and metrics. Removing the semantic embeddings reduced rating prediction performance on the Office dataset, produced mixed results on the Video Games dataset, and improved both MSE and MAE on the CDs & Vinyl dataset. The experiments therefore did not establish a consistent benefit or disadvantage of semantic aspect embeddings for rating prediction. In

contrast, removing the semantic aspect embeddings consistently improved both Hit@1 and Recall@1 for aspect prediction across all three datasets. In the original DRL, semantic aspect embeddings were incorporated into the edge encoder by mapping the sentiment-based aspect vector through the aspect embedding matrix, with the aim of encoding semantic information among aspects. The ablation results, however, did not show that this additional semantic representation consistently improved the evaluated tasks. One possible explanation for the aspect-prediction results is a mismatch between the semantic information encoded by this component and the exact-match evaluation protocol. For example, semantically related predictions such as "music" and "song" were treated as different aspects even though they could convey similar information. An alternative explanation is that the semantic representation itself introduced information that was not useful for predicting the specific aspects subsequently mentioned by users. The current experiments could not distinguish between these explanations. Further evaluation using semantic-aware metrics and human judgement would therefore be required to determine whether this component captures useful information that is not reflected by exact aspect matching.

A further qualification concerns the meaning of "explainability" in the reported results. The evaluation measured whether DRL could predict positive aspects that subsequently appeared in a user's review. These aspect predictions provided structured and human-readable information that could serve as the basis for explaining why an item matched the modelled user preferences. However, higher Hit@N or Recall@N did not directly demonstrate that users would perceive the resulting explanations as more useful, understandable, persuasive, or trustworthy. The reported results should therefore be interpreted as improvements in *aspect prediction as an operational measure of explanation*, rather than direct evidence of improved user-perceived explanation quality.

Taken together, the findings answered the research question in two parts. First, interaction-dependent user and item representations were learned by combining general collaborative representations with fine-grained aspect-sentiment information associated with the target interaction, allowing the representation of each entity to depend on its current counterpart. Second, within the evaluated datasets, these representations improved both rating prediction and positive-aspect prediction relative to the compared static approaches. The results therefore supported interaction-dependent representation learning as an effective strategy for the aspect-level recommendation

setting investigated in this study. At the same time, the ablation and evaluation results showed that the usefulness of individual representation components depended on the prediction objective and that improved automatic aspect prediction should not be equated with improved explanation quality from the user’s perspective.

4.6.1 Limitations and Future Work

The first limitation concerned the scope of the explanation task and its evaluation. DRL predicted only positive aspects and therefore provided supporting reasons for recommendations, but it did not model reasons why an item should not be recommended. Existing aspect-based recommendation research has shown that negative item performance on aspects valued by a user can also provide a basis for explaining disrecommendations (Zhang et al., 2014a). The datasets used in this study contained substantial negative aspect information, suggesting that the same general framework could be extended to predict both positive and negative aspects. Future work could therefore model these signals jointly to provide contrastive explanations that identify both the characteristics that support a recommendation and those that may conflict with the user’s preferences.

The current evaluation protocol imposed a second limitation. Positive aspects appearing in held-out reviews were treated as ground truth and evaluated through exact matching. This made the evaluation reproducible, but it did not recognise semantically equivalent aspect expressions and did not measure how users perceived the explanations. This limitation was particularly relevant to the semantic-ablation results, where removing semantic aspect embeddings improved the exact-match metrics. The experiments therefore did not establish whether the semantic component provided useful information that was not captured by the current evaluation protocol, or whether it was simply unnecessary for the aspect-prediction task. Future work should evaluate aspect predictions using semantic-aware matching in addition to exact matching and, more importantly, complement offline metrics with user studies. Recent research on explainable recommender systems has emphasised that user-centred evaluation is necessary for assessing properties such as usefulness, comprehensibility, persuasiveness, and trust, and that these effects can vary across user groups (Wardatzky et al., 2025). Such evaluation would help determine whether semantic representations produce explanations that users regard as equivalent or useful even when the predicted

wording differs from the review ground truth. If no benefit were observed under either semantic-aware or human evaluation, the semantic component could instead be removed from the model.

A third limitation concerned the construction of the aspect-sentiment data. Before adopting the final LLM-based extraction procedure, we performed a preliminary analysis using Sentires (Zhang et al., 2014b) on the original Amazon Video Games review corpus. Sentires identified many meaningful aspects, but the resulting vocabulary also exhibited substantial fragmentation among synonymous or morphologically related expressions, incomplete aspect terms, and sparse extraction at the review level. These observations motivated the use of an LLM-based procedure that could be prompted to infer contextually relevant aspects and sentiment polarities.

This preliminary analysis, however, was not a controlled comparison between Sentires and the LLM. It was conducted on the original, unprocessed corpus, whereas the final aspect-sentiment datasets were constructed through a different preprocessing pipeline. Moreover, no manually annotated aspect-sentiment ground truth was available to compare extraction precision, recall, semantic consistency, or implicit-aspect identification. The study therefore did not establish that LLM-based extraction was generally superior to Sentires. Future work should conduct a controlled comparison in which Sentires, LLM-based extraction, and more recent ABSA methods are applied to the same reviews and evaluated against manually annotated samples. Such evaluation should consider both intrinsic extraction quality and downstream effects on recommendation accuracy and aspect prediction. This would help separate improvements attributable to DRL from those attributable to the preprocessing method used to construct its aspect-sentiment inputs.

Beyond improving the evaluation of aspect extraction itself, LLMs could also be investigated as components of the explanation stage rather than being restricted to data preprocessing. DRL currently produced aspect-level outputs rather than complete natural-language explanations. Recent work has investigated the use of LLMs to generate user-facing recommendation explanations, although current research also distinguishes fluent recommendation justifications from faithful explanations of the underlying recommendation process (Said, 2025; Lin et al., 2025). A future extension could use the user and item aspect representations learned by DRL as explicit grounding evidence for an LLM, allowing it to verbalise the predicted aspects as personalised explanations. Grounding the generated text in the model’s predicted evidence would be important to reduce the risk of producing plausible but unsupported justifications.

A fourth limitation was the meaning of "dynamic" representation in the current model. DRL adapted user and item representations to the current interaction counterpart, but the model did not capture temporal changes in user preferences as new interactions occurred. This distinguishes interaction dependence from the temporal dynamics studied in sequential recommender systems. Recent sequential recommendation research has increasingly focused on modelling evolving short-term and long-term user interests from ordered interaction histories (Boka et al., 2024). A natural extension would therefore combine interaction-dependent representation learning with sequential modelling, allowing the representation used for a target user-item pair to depend both on the current counterpart and on the user's evolving interaction history. This would extend the current interaction-dependent formulation to incorporate temporal dependence as well.

The empirical scope of the study also limited the generalisability of the findings. Although three Amazon product categories with different characteristics were evaluated, all were drawn from the same e-commerce ecosystem. The experiments therefore did not establish whether the same interaction-dependent representation mechanism would provide similar benefits in domains with substantially different user behaviour, item characteristics, or review practices. Future work should validate DRL across additional domains and, where task-compatible implementations are available, compare it with more recent target-aware, context-aware, and LLM-enhanced recommendation approaches (Kim et al., 2024; Mateos and Bellogín, 2024). Such evaluation would provide stronger evidence about which aspects of DRL generalise beyond the current datasets.

Finally, the edge encoder produced predicted values corresponding to the dummy rating aspect in the user attention and item quality vectors, but these values were discarded in the final model. They may encode information related to a user's general rating tendency and an item's overall perceived quality. Future work could examine whether explicitly incorporating these outputs into rating prediction provides complementary information. This should be evaluated through controlled ablation rather than assumed to improve performance.

4.7 Conclusions

This study proposed DRL, an aspect-level recommendation model that learned interaction-dependent representations of users and items by combining general collaborative information with aspect-sentiment information associated with the current user-item interaction. In contrast to using a single fixed representation for every interaction, DRL allowed the representation of a user to depend on the target item and the representation of an item to depend on the target user. The experimental results showed that this design improved rating prediction and positive-aspect prediction relative to the compared baselines across the three evaluated datasets.

The main finding was that modelling the current user-item interaction explicitly provided useful information beyond general historical representations. The rating-based GNN contributed to both recommendation and aspect prediction, while the interaction-dependent layer enabled the model to relate the user's aspect preferences to the characteristics of the target item. The ablation study also showed that additional representation components were not uniformly beneficial. In particular, semantic aspect embeddings had mixed effects on rating prediction across datasets, while removing them consistently improved the exact-match aspect-prediction metrics. This finding highlighted the importance of evaluating representation components separately across different objectives rather than assuming that additional semantic information necessarily improves recommendation performance.

Overall, the findings showed one way in which user-item mutual influence could be incorporated into representation learning: general user and item representations were conditioned on interaction-specific aspect-sentiment evidence to construct representations for the target pair. Within the scope of the experiments, these representations supported more accurate rating and positive-aspect prediction. The latter provided a structured basis for aspect-level recommendation explanations, although the quality of these explanations from the user's perspective remains to be established through human-centred evaluation.

Finally, this study constructed three review datasets containing structured aspect-sentiment information using an LLM-based extraction procedure to support the aspect-level modelling and evaluation. Together, the proposed model and datasets provided a basis for further research on interaction-dependent representation learning and explainable recommendation.

Chapter 5

Discussion

Personalised recommender systems typically rely on representations of users and items learned from their historical interactions and, where available, additional sources of information. This thesis investigated how user and item representations could be learned from multiple forms of feedback, including implicit interactions, numerical ratings, and textual reviews, to improve recommendation accuracy and explainability. The overarching research question was: **How can user and item representations be learned from heterogeneous feedback to account for differences across feedback types and user-item interactions while supporting accurate and aspect-level explainable recommendation?**

To address this research question, this thesis developed a **polymorphic representation learning perspective**, in which the same entity could be represented differently according to the feedback type being modelled or the current user-item interaction. This perspective was investigated from two implementations:

- **Implementation 1: Implicit versus Explicit Feedback.** As discussed in Chapter 3, we proposed a GCN-based model, **Interdependent Embedding Learning (IEL)**, which learned **distinct yet interdependent** user and item representations from implicit and explicit feedback. IEL was designed to improve both ranking and rating prediction by allowing information to propagate within and across the two feedback-specific graphs.
- **Implementation 2: User-Item Mutual Influence.** In Chapter 4, we developed an aspect-level explainable recommendation model, **Dynamic Representation Learning (DRL)**, which learned **interaction-dependent** user and item representations according to their current interacting counterparts. The term "dynamic" in the model name refers to variation across user-item interactions rather than

temporal evolution. DRL was designed to improve rating prediction and aspect-level explanation by combining collaborative information with aspect-sentiment information extracted from textual reviews.

5.1 Key Findings of IEL

In the initial design of IEL, we incorporated an attention mechanism for neighbour aggregation, nonlinear interaction learning, and edge learning. These additional mechanisms did not improve ranking accuracy in the experiments. The final model therefore focused on node representation learning and adopted simpler components for intra-graph and inter-graph message passing, drawing on the designs of He et al. (2020) and Koren (2008a), respectively. An inner product was also used for interaction modelling. These results indicated that increasing architectural complexity did not necessarily improve the integration of heterogeneous feedback and that relatively simple propagation mechanisms could provide an effective basis for cross-feedback representation learning.

The evaluation of IEL showed that jointly modelling implicit and explicit feedback improved recommendation performance under the experimental setting, but the two feedback types contributed differently to the two recommendation objectives. Explicit feedback provided substantial additional information for ranking, whereas incorporating implicit feedback produced comparatively limited gains in rating prediction. This asymmetric effect was consistent with the observation of Chen and Peng (2018) that implicit signals can be too noisy to provide precise adjustments to rating estimates. A similar pattern was observed for the RnR baseline (Hadash et al., 2018), whose strongest rating prediction performance on both datasets was obtained when the weight assigned to its ranking objective was only 0.1. These findings suggested that heterogeneous feedback could provide complementary information, but that the usefulness of each feedback type depended on the prediction objective.

5.2 Key Findings of DRL

In the initial design of DRL, the learned interaction-dependent user and item representations were directly combined, for example through concatenation or element-wise multiplication, to predict ratings and review aspects. Although this design achieved

strong rating prediction performance, it performed less effectively on aspect prediction. The final model therefore first decoded a user-aspect attention vector and an item-aspect quality vector from the respective representations and then matched these vectors through element-wise multiplication. This decomposition substantially improved aspect prediction and provided an explicit connection between the learned representations and the aspect-level information used for recommendation explanation.

DRL outperformed the compared static baselines (Zhang et al., 2014a; Hou et al., 2019) in rating prediction and in the automatic aspect-prediction metrics used to evaluate explanation. The comparison with AARM (Guan et al., 2019) also suggested that incorporating sentiment associated with review aspects was useful for modelling fine-grained user preferences and item characteristics. However, the ablation experiments revealed an important qualification. Removing the semantic aspect embeddings improved the reported explanation metrics on all three datasets. Under the current experimental setting, the semantic component therefore did not provide an empirical advantage for aspect prediction. This result also exposed a limitation of the evaluation protocol, which relied on exact correspondence with aspects appearing in held-out reviews and could not assess whether semantically related but lexically different aspects would be perceived as equivalent by users.

Overall, the results supported the value of combining heterogeneous feedback and interaction-dependent representation learning under the evaluated settings. They also showed that the contribution of individual information sources and model components was task-dependent, and that improved automatic explanation metrics should not be interpreted as direct evidence of improved user-perceived explanation quality.

5.3 Addressing the Research Questions

IEL and DRL were developed to address the two sub-questions:

RQ1 How can implicit and explicit feedback be integrated to learn user and item representations that preserve feedback-specific information and support both item ranking and rating prediction?

RQ2 How can user and item representations be conditioned on each other to support both rating prediction and aspect-level recommendation explanation?

Regarding RQ1, IEL showed that feedback-specific user and item representations could be learned jointly through intra-graph and inter-graph information propagation. The resulting model improved both ranking and rating prediction relative to the compared baselines, although the benefits of the two feedback types were asymmetric. The findings therefore answered RQ1 by demonstrating that implicit and explicit feedback could be integrated through distinct yet connected representation spaces, while also showing that their contributions depended on the target recommendation task.

Regarding RQ2, DRL showed that user and item representations could be conditioned on each other by combining general collaborative representations with interaction-specific aspect information. This design improved rating prediction and aspect-prediction performance relative to the compared static models. The results therefore supported interaction-dependent representation learning as an effective strategy within the evaluated aspect-level recommendation setting. At the same time, because explanation quality was evaluated primarily through automatic aspect metrics and qualitative case studies, further user-centred evaluation would be required to determine whether these improvements translated into explanations that users found more useful, persuasive, or understandable.

Taken together, IEL and DRL demonstrated two implementations of the polymorphic representation learning perspective investigated in this thesis. IEL varied representations according to feedback type, whereas DRL adapted them according to the interacting counterpart. The overarching research question was therefore answered by showing that representation learning could explicitly account for heterogeneity in both feedback sources and current user-item interactions, rather than requiring the same representation to serve all recommendation settings.

5.4 Thesis Significance and Broader Impact

Classical collaborative filtering methods commonly associate each user and item with a general latent representation, whereas subsequent research has introduced richer alternatives, including multi-interest, disentangled, context-aware, and target-aware representations (Ma et al., 2019; Li et al., 2019). Against this broader research background, the contribution of this thesis was not simply the use of multiple or interaction-dependent representations. Instead, it formulated these variations within a common polymorphic representation perspective and investigated two complementary sources of representation variation. IEL modelled user and item representations separately

across implicit and explicit feedback while explicitly allowing information exchange between the corresponding representation spaces. DRL considered a different form of variation by adapting both user and item representations to the current interaction counterpart using fine-grained information derived from reviews. Together, the two studies provided a unified account of how representation learning could respond to heterogeneity in both feedback type and current user-item interaction.

A second contribution concerned the connection between representation learning and explainable recommendation. DRL incorporated aspect-level preference and item-quality information into the representation and prediction process, allowing recommended items to be associated with specific aspects that reflected the modelled relationship between user preferences and item characteristics. This provided a more inspectable basis for explanation than relying exclusively on latent user-item matching scores. Importantly, the experiments established improvements in aspect-prediction metrics rather than user-perceived explanation quality. The broader value of this approach therefore lies in providing structured, human-readable information that can support explanation generation, while its effects on user trust, satisfaction, and decision-making remain questions for future user-centred evaluation.

The research questions addressed in this thesis also remain relevant to current developments in recommender systems. Recent multi-behaviour recommendation research increasingly considers heterogeneous interaction signals and the dependencies between them, while multi-modal and LLM-based recommendation extends representation learning to richer textual, visual, and semantic information (Kim et al., 2026; Liu et al., 2024; Lin et al., 2025; Zhao et al., 2024b). The polymorphic perspective investigated in this thesis provides one way of reasoning about such settings: rather than forcing heterogeneous evidence into a single invariant representation, the representation used for a user or item can depend on the source and context of the available evidence. The experiments in this thesis were conducted in offline e-commerce settings, so they did not establish scalability or generalisability across domains. Nevertheless, the underlying representation principle provides a starting point for investigating more heterogeneous, multi-modal, and adaptive recommendation settings.

5.5 Limitations and Future Work

Several limitations constrained the conclusions that could be drawn from the experiments. First, the proposed models were evaluated primarily on Amazon product

datasets. Although multiple product categories were included, all datasets originated from the same e-commerce ecosystem, and the experiments therefore did not establish generalisability to substantially different recommendation domains. The thesis also did not evaluate system-level scalability through runtime, memory consumption, or large-scale deployment experiments. Future work should consequently examine the proposed representation learning perspective on larger datasets and in domains such as media, news, local services, or other platforms with different interaction patterns, while explicitly evaluating computational efficiency.

A second limitation concerned the modelling of implicit feedback. In the IEL experiments, implicit interactions were derived by binarising observed rating interactions rather than being collected as independent behavioural signals such as clicks, views, add-to-cart events, and purchases. The experiments therefore captured the distinction between binary and rating-based representations of interactions, but did not capture the different semantics of real user behaviours. Recent multi-behaviour recommendation research explicitly distinguishes behaviour-specific signals and studies how dependencies among them should be modelled (Kim et al., 2026). A natural extension of IEL would therefore be to learn separate but related representations for different behavioural types and to investigate when they should be combined, transferred, or kept distinct. Such data would also make it possible to extend the original proposal of combining IEL and DRL: aspect-level information could be used to investigate why the same user exhibited different behaviours towards an item, for example clicking or purchasing an item but later expressing dissatisfaction in a rating or review.

The interaction dependence studied by DRL also differed from temporal dynamics. DRL adapted a representation according to the current user-item counterpart, but it did not model how preferences evolved over time as new interactions arrived. Future work could extend polymorphic representation learning from interaction-dependent representations to representations that also adapt to behavioural sequence and temporal context. This direction would connect the framework to current multi-behaviour and sequential recommendation research, where both the type and order of interactions can provide information about changing user intent (Kim et al., 2026). Such an extension would also permit evaluation in online or continually updated settings rather than relying exclusively on representations learned from fixed historical data.

The evaluation of explainability presented another limitation. DRL was evaluated using automatic aspect-prediction metrics and qualitative case studies, but no user

study was conducted to assess whether the resulting explanations were understandable, useful, or persuasive to users. Future work should therefore combine automatic evaluation with human-centred assessment and should examine both positive and negative aspects when explaining recommendations or non-recommendations. Large Language Models provide an additional direction for this research. Existing studies have used LLMs as semantic encoders for user and item representation, as components of recommendation pipelines, and as generators of natural-language recommendation explanations (Zhao et al., 2024b; Lin et al., 2025; Said, 2025). In this thesis, the LLM was used only to extract aspect-sentiment information from reviews. Future work could investigate whether LLM-derived semantic representations complement collaborative representations and whether LLMs can generate personalised explanations grounded explicitly in the aspect-level evidence learned by DRL. Grounding would be particularly important because fluent LLM-generated justifications should not automatically be treated as faithful explanations of the recommendation model.

Finally, the current models relied predominantly on collaborative feedback and textual reviews. Incorporating item descriptions, images, and other metadata could provide complementary information when interaction histories are sparse and could reduce the dependence of representation learning on collaborative signals. Recent multimodal recommendation research has increasingly investigated the integration of textual, visual, and other content features for richer item representations and for sparse or cold-start settings (Liu et al., 2024). A future extension could therefore connect aspects extracted from reviews with corresponding item content, such as linking an aspect describing "storage" to product specifications or descriptions. This would allow the polymorphic framework to combine behavioural, evaluative, and content-based evidence and could provide a more informative representation for items with limited historical interactions.

Chapter 6

Conclusions

This thesis investigated how user and item representations can be learned from heterogeneous feedback when the information relevant to a recommendation varies across feedback types and interaction counterparts. The two studies were brought together under a **polymorphic representation learning** perspective, in which the same user or item does not necessarily need to be represented by a single invariant embedding. Instead, representations may differ according to the feedback being modelled or the particular user-item interaction being considered.

The first study addressed the integration of implicit and explicit feedback through **Interdependent Embedding Learning (IEL)**. IEL maintained separate user and item representations for the two feedback types and connected selected feedback-specific representations through GCN-based intra-graph and inter-graph message passing. The experimental results showed that modelling these representations as distinct but interdependent improved recommendation performance relative to the compared unified recommendation baselines. The benefits were strongest and most consistent for item ranking, whereas the contribution of the implicit-feedback view to rating prediction was more limited. These findings supported the view that implicit and explicit feedback can provide complementary information, but that their contributions need not be symmetric across recommendation objectives.

The second study addressed interaction-dependent representation learning through **Dynamic Representation Learning (DRL)**. DRL combined general collaborative representations with fine-grained aspect-sentiment information to construct an item-dependent user representation and a user-dependent item representation. Across the evaluated datasets, DRL achieved lower rating-prediction MSE than the compared baselines and improved positive-aspect prediction. These results provided empirical support for conditioning user and item representations on their current interaction counterpart. In this thesis, the term *dynamic* therefore refers to interaction-dependent representation

rather than temporal changes in user preferences or item characteristics.

Together, the two studies answered the research questions from two complementary perspectives. RQ1 examined how representations associated with implicit and explicit feedback could remain distinct while exchanging collaborative information. IEL addressed this problem through feedback-specific graphs and selected cross-feedback message-passing pathways. RQ2 examined how representations could vary according to the current user-item interaction. DRL addressed this problem by combining general collaborative information with interaction-specific aspect-sentiment information to construct interaction-dependent representations for rating and positive-aspect prediction.

The findings also clarify the role of explainability in this thesis. DRL produced structured aspect-level outputs that could be associated with the modelled preferences of a user and characteristics of an item. Higher Hit@N and Recall@N therefore indicated better prediction of positive aspects appearing in held-out reviews. These measures provided an operational evaluation of aspect-level explanation, but they did not directly establish that users would perceive the resulting explanations as more useful, understandable, persuasive, or trustworthy. This distinction is important when interpreting the contribution of DRL to explainable recommendation.

The main contribution of this thesis is therefore not the claim that a single representation is universally inadequate, but the demonstration that allowing representations to vary in a controlled manner can be beneficial when the available information has different meanings or when different information becomes relevant for different interaction counterparts. IEL demonstrated this principle across feedback types, while DRL demonstrated it across user-item interaction contexts. The thesis also constructed three derived Amazon review datasets containing LLM-extracted aspect-sentiment information to support aspect-level recommendation research.

Overall, this thesis provides empirical evidence that feedback-dependent and interaction-dependent representation learning offer useful ways to model heterogeneous user feedback in recommender systems. Rather than collapsing different signals or user-item interactions into a single representation, the proposed approaches preserve their distinctions while allowing relevant information to be integrated for prediction. This perspective provides a foundation for further research on representation learning for accurate and explainable recommendation.

Bibliography

- Abdollahi, Behnoush and Olfa Nasraoui (2016). “Explainable matrix factorization for collaborative filtering”. In: *Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 5–6.
- Adomavicius, Gediminas and Alexander Tuzhilin (2005). “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”. In: *IEEE transactions on knowledge and data engineering* 17.6, pp. 734–749.
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations (ICLR)*. URL: <https://arxiv.org/abs/1409.0473>.
- Bai, Yinhao, Zhixin Han, Yuhua Zhao, Hang Gao, Zhuowei Zhang, Xunzhi Wang, and Mengting Hu (2024). “Is Compound Aspect-Based Sentiment Analysis Addressed by LLMs?” In: *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7836–7861.
- Balabanović, Marko and Yoav Shoham (1997). “Fab: content-based, collaborative recommendation”. In: *Communications of the ACM* 40.3, pp. 66–72.
- Bao, Zhuolan, Wenwen Li, Pengzhen Yin, and Michael Chau (2021). “Examining the impact of review tag function on product evaluation and information perception of popular products”. In: *Information Systems and e-Business Management* 19.2, pp. 517–539.
- Bauman, Konstantin, Bing Liu, and Alexander Tuzhilin (2017). “Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 717–725.
- Bell, Robert M and Yehuda Koren (2007). “Scalable collaborative filtering with jointly derived neighborhood interpolation weights”. In: *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, pp. 43–52.
- Bellogín, Alejandro, Pablo Castells, and Iván Cantador (2017). “Statistical biases in information retrieval metrics for recommender systems”. In: *Information Retrieval Journal* 20, pp. 606–634.

- Boka, Tesfaye Fenta, Zhendong Niu, and Rama Bastola Neupane (2024). "A survey of sequential recommendation systems: Techniques, evaluation, and future directions". In: *Information Systems* 125, p. 102427.
- Breese, John S., David Heckerman, and Carl Kadie (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI98)*, pp. 43–52.
- Brody, Shaked, Uri Alon, and Eran Yahav (2021). "How attentive are graph attention networks?" In: *arXiv preprint arXiv:2105.14491*.
- Cao, Qimeng, Qing Yin, Yunya Song, Zhihua Wang, Yujun Chen, Richard Yi Da Xu, and Xian Yang (2023). "RTANet: Recommendation target-aware network embedding". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 17, pp. 84–94.
- Catherine, Rose and William Cohen (2017). "Transnets: Learning to transform for recommendation". In: *Proceedings of the eleventh ACM conference on recommender systems*, pp. 288–296.
- Chen, Chong, Min Zhang, Yiqun Liu, and Shaoping Ma (2018). "Neural attentional rating regression with review-level explanations". In: *Proceedings of the 2018 world wide web conference*, pp. 1583–1592.
- Chen, Jingyuan, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua (2017). "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention". In: *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 335–344.
- Chen, Lei, Le Wu, Richang Hong, Kun Zhang, and Meng Wang (2020a). "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 01, pp. 27–34.
- Chen, Shulong and Yuxing Peng (2018). "Matrix factorization for recommendation with explicit and implicit feedback". In: *Knowledge-Based Systems* 158, pp. 109–117.
- Chen, Tong, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang (2020b). "Try this instead: Personalized and interpretable substitute recommendation". In: *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 891–900.
- Cremonesi, Paolo, Yehuda Koren, and Roberto Turrin (2010). "Performance of recommender algorithms on top-n recommendation tasks". In: *Proceedings of the fourth ACM conference on Recommender systems*, pp. 39–46.

- De Campos, Luis M, Juan M Fernández-Luna, and Juan F Huete (2024). "An explainable content-based approach for recommender systems: a case study in journal recommendation for paper submission". In: *User Modeling and User-Adapted Interaction*, pp. 1–35.
- Deng, Zhi-Hong, Ling Huang, Chang-Dong Wang, Jian-Huang Lai, and S Yu Philip (2019). "Deepcf: A unified framework of representation learning and matching function learning in recommender system". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 61–68.
- Fu, Mingsheng, Hong Qu, Dagmawi Moges, and Li Lu (2018). "Attention based collaborative filtering". In: *Neurocomputing* 311, pp. 88–98.
- Gao, Jingyue, Xiting Wang, Yasha Wang, and Xing Xie (2019). "Explainable recommendation through attentive multi-view learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 3622–3629.
- Graves, Alex (2013). "Generating sequences with recurrent neural networks". In: *arXiv preprint arXiv:1308.0850*.
- Guan, Xinyu, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua (2019). "Attentive aspect modeling for review-aware recommendation". In: *ACM Transactions on Information Systems (TOIS)* 37.3, pp. 1–27.
- Guo, Huifeng, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He (2017). "DeepFM: a factorization-machine based neural network for CTR prediction". In: *arXiv preprint arXiv:1703.04247*.
- Hadash, Guy, Oren Sar Shalom, and Rita Osadchy (2018). "Rank and rate: multi-task learning for recommender systems". In: *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 451–454.
- Hasan, Emrul, Mizanur Rahman, Chen Ding, Jimmy Xiangji Huang, and Shaina Raza (2025). "Review-based recommender systems: a survey of approaches, challenges and future perspectives". In: *ACM Computing Surveys* 58.1, pp. 1–41.
- He, Xiangnan, Tao Chen, Min-Yen Kan, and Xiao Chen (2015). "Trirank: Review-aware explainable recommendation by modeling aspects". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 1661–1670.
- He, Xiangnan, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang (2018a). "Outer Product-based Neural Collaborative Filtering". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2227–2233.

- He, Xiangnan, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang (2020). "Lightgcn: Simplifying and powering graph convolution network for recommendation". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648.
- He, Xiangnan, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua (2018b). "NAIS: Neural attentive item similarity model for recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* 30.12, pp. 2354–2366.
- He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua (2017). "Neural collaborative filtering". In: *Proceedings of the 26th international conference on world wide web*, pp. 173–182.
- Herlocker, Jonathan L, Joseph A Konstan, and John Riedl (2000). "Explaining collaborative filtering recommendations". In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 241–250.
- Hou, Yunfeng, Ning Yang, Yi Wu, and Philip S Yu (2019). "Explainable recommendation with fusion of aspect information". In: *World Wide Web* 22, pp. 221–240.
- Hu, Yifan, Yehuda Koren, and Chris Volinsky (2008). "Collaborative filtering for implicit feedback datasets". In: *2008 Eighth IEEE international conference on data mining*. Ieee, pp. 263–272.
- Hua, Yan Cathy, Paul Denny, Jörg Wicker, and Katerina Taskova (2024). "A systematic review of aspect-based sentiment analysis: domains, methods, and trends". In: *Artificial Intelligence Review* 57.11, p. 296.
- Hulstijn, Joris, Igor Tchappi, Amro Najjar, and Reyhan Aydoğan (2023). "Metrics for evaluating explainable recommender systems". In: *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*. Springer, pp. 212–230.
- Jadidinejad, Amir H, Craig Macdonald, and Iadh Ounis (2019). "Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks". In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 149–156.
- Jawaheer, Gawesh, Martin Szomszor, and Patty Kostkova (2010). "Comparison of implicit and explicit feedback from an online music recommendation service". In: *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, pp. 47–51.

- Kabbur, Santosh, Xia Ning, and George Karypis (2013). “Fism: factored item similarity models for top-n recommender systems”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 659–667.
- Kim, Kyungho, Sunwoo Kim, Geon Lee, Jinhong Jung, and Kijung Shin (2026). “A comprehensive survey on multi-behavior recommender systems: extended taxonomy and recent advances”. In: *International Journal of Data Science and Analytics* 22.1, p. 40.
- Kim, Yungi, Taeri Kim, Won-Yong Shin, and Sang-Wook Kim (2024). “MONET: Modality-embracing graph convolutional network and target-aware attention for multimedia recommendation”. In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 332–340.
- Kipf, Thomas N and Max Welling (2016). “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907*.
- Koren, Yehuda (2008a). “Factorization meets the neighborhood: a multifaceted collaborative filtering model”. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434.
- (2008b). “Tutorial on recent progress in collaborative filtering”. In: *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 333–334.
- Koren, Yehuda, Robert Bell, and Chris Volinsky (2009). “Matrix factorization techniques for recommender systems”. In: *Computer* 42.8, pp. 30–37.
- Koren, Yehuda, Steffen Rendle, and Robert Bell (2021). “Advances in collaborative filtering”. In: *Recommender systems handbook*, pp. 91–142.
- Lau, Jey Han and Timothy Baldwin (2016). “An empirical evaluation of doc2vec with practical insights into document embedding generation”. In: *arXiv preprint arXiv:1607.05368*.
- Lee, Daniel D and H Sebastian Seung (1999). “Learning the parts of objects by non-negative matrix factorization”. In: *nature* 401.6755, pp. 788–791.
- Li, Chao, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee (2019). “Multi-interest network with dynamic routing for recommendation at Tmall”. In: *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2615–2623.
- Li, Lei, Yongfeng Zhang, and Li Chen (2023). “Personalized prompt learning for explainable recommendation”. In: *ACM Transactions on Information Systems* 41.4, pp. 1–26.

- Li, Sheng and Handong Zhao (2021). "A survey on representation learning for user modeling". In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 4997–5003.
- Liang, Dawen, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara (2018). "Variational autoencoders for collaborative filtering". In: *Proceedings of the 2018 world wide web conference*, pp. 689–698.
- Lin, Jianghao, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, et al. (2025). "How can recommender systems benefit from large language models: A survey". In: *ACM Transactions on Information Systems* 43.2, pp. 1–47.
- Linden, Greg, Brent Smith, and Jeremy York (2003). "Amazon. com recommendations: Item-to-item collaborative filtering". In: *IEEE Internet computing* 7.1, pp. 76–80.
- Liu, Huafeng, Liping Jing, Jingxuan Wen, Pengyu Xu, Jiaqi Wang, Jian Yu, and Michael K Ng (2021). "Interpretable deep generative recommendation models". In: *Journal of Machine Learning Research* 22.202, pp. 1–54.
- Liu, Huazhen, Wei Wang, Yihan Zhang, Renqian Gu, and Yaqi Hao (2022). "Neural matrix factorization recommendation for user preference prediction based on explicit and implicit feedback". In: *Computational Intelligence and Neuroscience* 2022.1, p. 9593957.
- Liu, Jian, Chuan Shi, Binbin Hu, Shenghua Liu, and Philip S Yu (2017). "Personalized ranking recommendation via integrating multiple feedbacks". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 131–143.
- Liu, Nathan N, Evan W Xiang, Min Zhao, and Qiang Yang (2010). "Unifying explicit and implicit feedback for collaborative filtering". In: *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1445–1448.
- Liu, Qidong, Jiaxi Hu, Yutian Xiao, Xiangyu Zhao, Jingtong Gao, Wanyu Wang, Qing Li, and Jiliang Tang (2024). "Multimodal recommender systems: A survey". In: *ACM Computing Surveys* 57.2, pp. 1–17.
- Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro (2011). "Content-based recommender systems: State of the art and trends". In: *Recommender systems handbook*, pp. 73–105.
- Luo, Huanrui, Ning Yang, and S Yu Philip (2019). "Hybrid deep embedding for recommendations with dynamic aspect-level explanations". In: *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 870–879.

- Lv, Yanxia, Ying Zheng, Fangna Wei, Cong Wang, and Cuirong Wang (2020). "AICF: Attention-based item collaborative filtering". In: *Advanced Engineering Informatics* 44, p. 101090.
- Ma, Jianxin, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu (2019). "Learning disentangled representations for recommendation". In: *Advances in neural information processing systems* 32.
- Ma, Qiyao, Xubin Ren, and Chao Huang (2024). "Xrec: Large language models for explainable recommendation". In: *Findings of the association for computational linguistics: EMNLP 2024*, pp. 391–402.
- Ma, Ting, Longtao Huang, Qianqian Lu, and Songlin Hu (2023). "Kr-gcn: Knowledge-aware reasoning with graph convolution network for explainable recommendation". In: *ACM Transactions on Information Systems* 41.1, pp. 1–27.
- Mandal, Supriyo and Abyayananda Maiti (2020). "Explicit feedback meet with implicit feedback in GPMF: a generalized probabilistic matrix factorization model for recommendation". In: *Applied Intelligence* 50.6, pp. 1955–1978.
- Mateos, Pablo and Alejandro Bellogín (2024). "A systematic literature review of recent advances on context-aware recommender systems". In: *Artificial Intelligence Review* 58.1, p. 20.
- McAuley, Julian and Jure Leskovec (2013). "Hidden factors and hidden topics: understanding rating dimensions with review text". In: *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172.
- McMahan, H Brendan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. (2013). "Ad click prediction: a view from the trenches". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1222–1230.
- Meng, Wenjing, Deqing Yang, and Yanghua Xiao (2020). "Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation". In: *Proceedings of the 43rd international ACM SIGIR conference on research and development in Information Retrieval*, pp. 1091–1100.
- Mikolov, Tomas, Edouard Grave, Piotr Bojanowski, Christian Puhresch, and Armand Joulin (2018). "Advances in Pre-Training Distributed Word Representations". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26.
- Mnih, Andriy and Ruslan Salakhutdinov (2008). "Probabilistic Matrix Factorization". In: *Advances in Neural Information Processing Systems*. Vol. 20, pp. 1257–1264. URL: <https://papers.nips.cc/paper/3208-probabilistic-matrix-factorization>.
- Ni, Jianmo, Jiacheng Li, and Julian McAuley (2019). "Justifying recommendations using distantly-labeled reviews and fine-grained aspects". In: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 188–197.
- Pan, Deng, Xiangrui Li, Xin Li, and Dongxiao Zhu (2020). "Explainable recommendation via interpretable feature mapping and evaluation of explainability". In: *arXiv preprint arXiv:2007.06133*.
- Pan, Sicheng, Dongsheng Li, Hansu Gu, Tun Lu, Xufang Luo, and Ning Gu (2022). "Accurate and explainable recommendation via review rationalization". In: *Proceedings of the ACM Web Conference 2022*, pp. 3092–3101.
- Pan, Weike, Shanchuan Xia, Zhuode Liu, Xiaogang Peng, and Zhong Ming (2016). "Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks". In: *Information Sciences* 332, pp. 84–93.
- Paterek, Arkadiusz (2007). "Improving regularized singular value decomposition for collaborative filtering". In: *Proceedings of KDD cup and workshop*. Vol. 2007. 2007, pp. 5–8.
- Pazzani, Michael J. and Daniel Billsus (2007). "Content-Based Recommendation Systems". In: *The Adaptive Web: Methods and Strategies of Web Personalization*. Vol. 4321. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 325–341.
- Peng, Furong, Fujin Liao, Xuan Lu, Jianxing Zheng, and Ru Li (2025). "Revisiting explicit recommendation with DC-GCN: divide-and-conquer graph convolution network". In: *Information Systems* 130, p. 102513.
- Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.
- Raza, Shaina, Mizanur Rahman, Safiullah Kamawal, Armin Toroghi, Ananya Raval, Farshad Navah, and Amirmohammad Kazemeini (2024). "A comprehensive review of recommender systems: Transitioning from theory to practice". In: *arXiv preprint arXiv:2407.13699*.

- Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme (2009). "BPR: Bayesian personalized ranking from implicit feedback". In: *UAI*, pp. 452–461.
- Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl (1994). "Grouplens: An open architecture for collaborative filtering of netnews". In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186.
- Roy, Deepjyoti and Mala Dutta (2022). "A systematic review and research perspective on recommender systems". In: *Journal of Big Data* 9.1, p. 59.
- Sachdeva, Noveen and Julian McAuley (2020). "How useful are reviews for recommendation? A critical review and potential improvements". In: *proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1845–1848.
- Said, Alan (2025). "On explaining recommendations with Large Language Models: a review". In: *Frontiers in Big Data* 7, p. 1505284.
- Salton, Gerard and Christopher Buckley (1988). "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5, pp. 513–523.
- Sedhain, Suvash, Aditya Krishna Menon, Scott Sanner, and Lexing Xie (2015). "Autorec: Autoencoders meet collaborative filtering". In: *Proceedings of the 24th international conference on World Wide Web*, pp. 111–112.
- Seo, Sungyong, Jing Huang, Hao Yang, and Yan Liu (2017). "Interpretable convolutional neural networks with dual local and global attention for review rating prediction". In: *Proceedings of the eleventh ACM conference on recommender systems*, pp. 297–305.
- Shi, Yue, Martha Larson, and Alan Hanjalic (2013). "Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation". In: *Information Sciences* 229, pp. 29–39.
- Shuai, Jie, Le Wu, Kun Zhang, Peijie Sun, Richang Hong, and Meng Wang (2023). "Topic-enhanced graph neural networks for extraction-based explainable recommendation". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1188–1197.
- Steck, Harald (2010). "Training and testing of recommender systems on data missing not at random". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 713–722.

- Steck, Harald (2013). "Evaluation of recommendations: rating-prediction and ranking". In: *Proceedings of the 7th ACM conference on Recommender systems*, pp. 213–220.
- Su, Xiaoyuan and Taghi M. Khoshgoftaar (2009). "A Survey of Collaborative Filtering Techniques". In: *Advances in Artificial Intelligence* 2009, 4:2–4:2.
- Sun, Peijie, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang (2020). "Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation". In: *Proceedings of The Web Conference 2020*, pp. 837–847.
- Sun, Peijie, Le Wu, Kun Zhang, Yu Su, and Meng Wang (2021). "An unsupervised aspect-aware recommendation model with explanation text generation". In: *ACM Transactions on Information Systems (TOIS)* 40.3, pp. 1–29.
- Tan, Juntao, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang (2021). "Counterfactual explainable recommendation". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1784–1793.
- Tintarev, Nava (2007). "Explanations of recommendations". In: *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 203–206.
- Tintarev, Nava and Judith Masthoff (2012). "Beyond explaining single item recommendations". In: *Recommender Systems Handbook*. Springer, pp. 711–756.
- Tran, Quyen, Lam Tran, Linh Chu Hai, Ngo Van Linh, and Khoat Than (2022). "From Implicit to Explicit feedback: A deep neural network for modeling sequential behaviours and long-short term preferences of online users". In: *Neurocomputing* 479, pp. 89–105.
- Turney, Peter D and Patrick Pantel (2010). "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37, pp. 141–188.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.
- Velickovic, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. (2018). "Graph attention networks". In: *Proceedings of 6th international conference on learning representations*, pp. 1–12.
- Vig, Jesse, Shilad Sen, and John Riedl (2009). "Tagsplanations: explaining recommendations using tags". In: *Proceedings of the 14th international conference on Intelligent user interfaces*, pp. 47–56.

- Wan, Mengting and Julian McAuley (2018). "Item recommendation on monotonic behavior chains". In: *Proceedings of the 12th ACM conference on recommender systems*, pp. 86–94.
- Wang, Jun, Arjen P De Vries, and Marcel JT Reinders (2006). "Unifying user-based and item-based collaborative filtering approaches by similarity fusion". In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 501–508.
- Wang, Ke, Yanmin Zhu, Haobing Liu, Tianzi Zang, and Chunyang Wang (2023). "Learning aspect-aware high-order representations from ratings and reviews for recommendation". In: *ACM Transactions on Knowledge Discovery from Data* 17.1, pp. 1–22.
- Wang, Nan, Hongning Wang, Yiling Jia, and Yue Yin (2018). "Explainable recommendation via multi-task learning in opinionated text data". In: *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 165–174.
- Wang, Xi, Iadh Ounis, and Craig Macdonald (2021). "Leveraging review properties for effective recommendation". In: *Proceedings of the Web Conference 2021*, pp. 2209–2219.
- Wang, Xiang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua (2019). "Neural graph collaborative filtering". In: *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 165–174.
- Wardatzky, Kathrin, Oana Inel, Luca Rossetto, and Abraham Bernstein (2025). "Whom do explanations serve? a systematic literature survey of user characteristics in explainable recommender systems evaluation". In: *ACM Transactions on Recommender Systems* 3.4, pp. 1–35.
- Wu, Le, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang (2023). "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* 35.5, pp. 4425–4445.
- Wu, Shiwen, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui (2022). "Graph neural networks in recommender systems: a survey". In: *ACM Computing Surveys* 55.5, pp. 1–37.
- Wu, Yao, Christopher DuBois, Alice X Zheng, and Martin Ester (2016). "Collaborative denoising auto-encoders for top-n recommender systems". In: *Proceedings of the ninth ACM international conference on web search and data mining*, pp. 153–162.

- Xie, Ruobing, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin (2020). "Deep Feedback Network for Recommendation". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, pp. 2519–2525.
- Xiong, Kun, Wenwen Ye, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, Binbin Hu, Zhiqiang Zhang, and Jun Zhou (2021). "Counterfactual review-based recommendation". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2231–2240.
- Xu, Jingcao, Chaokun Wang, Cheng Wu, Yang Song, Kai Zheng, Xiaowei Wang, Changping Wang, Guorui Zhou, and Kun Gai (2023). "Multi-behavior self-supervised learning for recommendation". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 496–505.
- Xue, Feng, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong (2019). "Deep item-based collaborative filtering for top-n recommendation". In: *ACM Transactions on Information Systems (TOIS)* 37.3, pp. 1–25.
- Xue, Hong-Jian, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen (2017). "Deep matrix factorization models for recommender systems." In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 17, pp. 3203–3209.
- Yang, Mengyuan, Mengying Zhu, Yan Wang, Linxun Chen, Yilei Zhao, Xiuyuan Wang, Bing Han, Xiaolin Zheng, and Jianwei Yin (2024). "Fine-tuning large language model based explainable recommendation with explainable quality reward". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 8, pp. 9250–9259.
- Yuan, Jiahao, Wendi Ji, Dell Zhang, Jinwei Pan, and Xiaoling Wang (2022). "Micro-behavior encoding for session-based recommendation". In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, pp. 2886–2899.
- Zhang, Chenyan, Shan Xue, Jing Li, Jia Wu, Bo Du, Donghua Liu, and Jun Chang (2023a). "Multi-aspect enhanced graph neural networks for recommendation". In: *Neural Networks* 157, pp. 90–102.
- Zhang, Jiani, Xingjian Shi, Shenglin Zhao, and Irwin King (2019a). "Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems". In: *arXiv preprint arXiv:1905.13129*.
- Zhang, Shuai, Lina Yao, Aixin Sun, and Yi Tay (2019b). "Deep learning based recommender system: A survey and new perspectives". In: *ACM computing surveys (CSUR)* 52.1, pp. 1–38.

- Zhang, Yi, Yiwen Zhang, Dengcheng Yan, Qiang He, and Yun Yang (2024). "NIE-GCN: Neighbor Item Embedding-Aware Graph Convolutional Network for Recommendation". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
- Zhang, Yijia, Wanli Zuo, Zhenkun Shi, and Binod Kumar Adhikari (2023b). "Integrating reviews and ratings into graph neural networks for rating prediction". In: *Journal of Ambient Intelligence and Humanized Computing* 14.7, pp. 8703–8723.
- Zhang, Yongfeng and Xu Chen (2020). "Explainable recommendation: A survey and new perspectives". In: *Foundations and Trends® in Information Retrieval* 14.1, pp. 1–101.
- Zhang, Yongfeng, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma (2014a). "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 83–92.
- Zhang, Yongfeng, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma (2014b). "Do users rate or review? Boost phrase-level sentiment labeling with review-level sentiment classification". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 1027–1030.
- Zhang, Yu, Peter Tiño, Aleš Leonardis, and Ke Tang (2021). "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5, pp. 726–742.
- Zhao, Yu, Qiang Xu, Ying Zou, and Wei Li (2023). "Modeling user reviews through bayesian graph attention networks for recommendation". In: *ACM Transactions on Information Systems* 41.3, pp. 1–29.
- Zhao, Zihan, Xiangrong Tong, Yingjie Wang, and Qiang Zhang (2024a). "Multi-behavior contrastive learning with graph neural networks for recommendation". In: *Knowledge-Based Systems* 300, p. 112221.
- Zhao, Zihuai, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. (2024b). "Recommender systems in the era of large language models (llms)". In: *IEEE Transactions on Knowledge and Data Engineering* 36.11, pp. 6889–6907.
- Zheng, Lei, Vahid Noroozi, and Philip S Yu (2017). "Joint deep modeling of users and items using reviews for recommendation". In: *Proceedings of the tenth ACM international conference on web search and data mining*, pp. 425–434.

- Zhou, Guorui, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai (2018a). “Deep interest network for click-through rate prediction”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1059–1068.
- Zhou, Meizi, Zhuoye Ding, Jiliang Tang, and Dawei Yin (2018b). “Micro behaviors: A new perspective in e-commerce recommender systems”. In: *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 727–735.